

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

PHP BĀZĒTA INTERNETA VEIKALA IZSTRĀDE UN
ATBILSTOŠO IETVARU IZPĒTE

MAGISTRA DARBS

Autore: **Viktorija Šijenoka**

Stud. apl. Nr. vs10001

Darba vadītājs: Dr. Datorzinātņu doktors, asoc.prof. Edgars Celms

RĪGA 2012

ANOTĀCIJA

Viktorijas Šijenokas maģistra darba mērķis ir *PHP* ietvaru izpētīšana un auto piederumu interneta veikala izveidošana. Mērķa sasniegšanai tika izvirzīti un atrisināti sekojoši uzdevumi: auto piederumu interneta veikala prasību definēšana, izvēlēto *PHP* ietvaru analīze un izpētīšana, labāka un piemērotāka ietvara izvēle, veikala izstrādāšana ar izvēlēto ietvaru.

Darba gaitā tika izpētīti pieci ietvari, kas tika salīdzināti pēc vienādiem kritērijiem. Vērtēšanai tika izmantota teorētiska informācija no izstrādātāju oficiālajiem avotiem un rezultāti iegūti no praktiskās pielietošanas.

Maģistra darba rezultātā tika izvēlēts atbilstošs ietvars un ar to palīdzību tika izstrādāts auto piederumu interneta veikals.

Atslēgvārdi: *PHP*, ietvars, e-komercija.

ANNOTATION

Development of PHP based internet shop and examination of appropriate frameworks.

The aim of Victoria Shienok master project is research of PHP frameworks and development of auto accessories internet shop. To reach the goal there was defined and solved a set of tasks: definition of the internet shop functional tasks, research and comparison of selected frameworks, choice of the appropriate framework and internet shop development.

During the task there was researched and compared five frameworks. Comparison was made using theoretical information taken from the official framework developer's web sites and obtained practical knowledge.

In the result of master project there was selected appropriate framework and developed auto accessories internet shop.

Keywords: PHP, frameworks, e-commerce.

AUTOREFERĀTS

Maģistra darbs ir veltīts *PHP* ietvaru izpētīšanai un salīdzināšanai, ar mērķi izvēlēties atbilstošu variantu auto piederumu interneta veikala izveidošanai. Darba sākumā ir veikts ieskats elektroniskajā komercijā, lai saprastu kādas priekšrocības interneta veikals var sniegt lietotājiem. Pirms ietvaru izpētīšanas tiek izvirzītas funkcionālās prasības, kuras ir svarīgas šāda veida projektam. Pēc tam tiek definēti kritēriji, pēc kuriem tiks salīdzināti ietvari.

Darba otrā daļa ir veltīta piecu *PHP* ietvaru pētīšanai. Šim nolūkam tika izvēlēti tādi ietvari kā: *CodeIgniter*, *CakePHP*, *Symfony*, *Zend Framework* un *Yii Framework*. Sākumā tiek apskatīta ietvaru izstrādātāju sniegtā informācija, kas paskaidro ietvara iespējas un prasības. Pēc teorētiskās informācijas apskata sākas pētīšanas praktiskā daļa. Katrs ietvars pēc kārtas tiek ieinstalēts autores datorā ar mērķi apskatīt teorijā aprakstīto ar reālām darbībām. Ietvari tiek salīdzināti pēc vienādiem kritērijiem, ar tiem tiek veiktas vienādas darbības. Kritēriji ir: uzstādīšana, programmu izstrādāšana, ātrums, drošība, dokumentācijas pilnīgums un citi, projektam aktuāli paplašinājumi un moduļi. Darba gaitā, ar katru ietvaru tiek izstrādātas divas programmas, kas parāda ietvaru būtību un darbību. Katrai programmai tiek izmērīts ātrums, iegūtie rezultāti tiek salīdzināti ar *Yii* komandas veiktajiem ātruma testiem. Pētīšanas rezultātā tiek izveidota salīdzinājuma tabula un uz to pamata autore izvēlās interneta veikala izstrādāšanai atbilstīgu ietvaru.

Maģistra darba rezultātā tika izstrādāts auto piederumu interneta veikals. Darba trešajā daļā ir aprakstīti tā izstrādāšanai izmantotie līdzekļi, paplašinājumi, ka arī ietvara sīkākas pētīšanas rezultātā, gūtās zināšanas.

Darba beigās ir izstrādāti secinājumi un priekšlikumi.

SATURA RĀDĪTĀJS

| | |
|------------------------------------------------|----------------------------------------|
| ABSTRACT | ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА. |
| ANOTĀCIJA | 4 |
| ANNOTATION | 4 |
| AUTOREFERĀTS..... | 4 |
| SATURA RĀDĪTĀJS..... | 5 |
| APZĪMĒJUMU SARAKSTS | 7 |
| 1. IEVADS | 8 |
| 2. AUTO PIEDERUMU VEIKALA APRAKSTS | 10 |
| 1.1. Interneta veikala priekšrocības. | 10 |
| 1.2. Prasību formulēšana..... | 12 |
| 1.2.1. Apraksts | 12 |
| 1.2.2. Struktūra..... | 16 |
| 3. IETVARU PĒTĪŠANA | 18 |
| 2.1. Pētīšanas kritēriju apraksts..... | 18 |
| 2.2. Ietvaru pētījuma rezultātu apraksts | 20 |
| 2.2.1. Ietvaru apraksts un prasības | 20 |
| 2.2.2. Uzstādīšana | 25 |
| 2.2.3. Programmu izstrādāšana | 32 |
| 2.2.4. Ātrums..... | 51 |
| 2.2.5. Drošība | 53 |
| 2.2.6. Dokumentācija | 58 |
| 2.2.7. Citi..... | 60 |
| 2.2.8. Vērtējumu tabula..... | 64 |
| 4. INTERNETA VEIKALA IZSTRĀDĀŠANA | 66 |
| 3.1. Apraksts | 66 |
| 3.2. Administratīvās daļas izstrādāšana | 67 |
| 3.3. Lietotāja daļas izstrādāšana..... | 70 |
| SECINĀJUMI UN PRIEKŠLIKUMI..... | 76 |

IZMANTOTĀ LITERATŪRA..... 78

APZĪMĒJUMU SARAKSTS

PHP – Hypertext Preprocessors – Hiperteksta priekšprocesors (no angļu val.). Skriptu valoda, kura sākotnēji bija paredzēta servera puses lietojumos dinamisku tīmekļa lapu ģenerēšanai.

SQL – Structured Query Language – strukturēta vaicājumu valoda (no angļu val.) , kas paredzēta datu manipulēšanai relāciju datubāžu pārvaldības sistēmās CMS – Content management system – satura pārvaldības sistēmas.

MVC – Model-View-Controller – Modelis-Skats-Kontrolleris (no angļu val.), projektēšanas šablons, kas sadala sistēmu trijās daļās - datu ievads, datu apstrāde, datu attēlojums kur katra daļa risina atsevišķu uzdevumu.

HTTP – HyperText Transfer Protocol – Hiperteksta transporta protokols (no angļu val.), protokols, kas paredzēts datu apmaiņai starp tīmekļa serveriem un pārlūkprogrammām.

ApacheBench –programa *HTTP* web servera veiktspējas ātruma mērīšanai. Ietilpst standarta Apache *HTTP* servera pakotnē.

ActiveRecord, Doctrine – ORM (Object Role Model) tehnoloģija, kas sasaista datu bāzi ar programmu.

CRUD – create retrieve update delete – izveidot ielādēt atjaunot izdzēst (no angļu val.).

APC – programma, kas paātrina PHP scenārijus.

HTML – HyperText Markup Language – Hiperteksta iezīmēšanas valoda (no angļu val.), valoda, kas ir izstrādāta tīmekļa vietņu un citas pārlūkprogrammā attēlojamas informācijas glabāšanai.

XSS – Cross Site Scripting – uzbrukuma veids tīmekļa vietnēm.

Captcha – ir tehnoloģija, kas pārbauda, vai sistēmas lietotājs ir cilvēks vai mašīna.

CSRF – Cross Site Request Forgery – uzbrukums, kas ir vērsts uz mājas lapas apmeklētājiem.

1.IEVADS

Maģistra darbs ir veltīts *PHP* ietvaru izpētīšanai un salīdzināšanai. Pamatojoties uz iegūtiem rezultātiem, darba gaitā tika izvēlēts labākais ietvars un izveidots auto piederumu interneta veikals. Maģistra darba mērķis ir *PHP* ietvaru izpētīšana un autopiederumu interneta veikala izstrādāšana. Lai sasniegtu mērķi darba autoresm ir jārisina sekojoši uzdevumi:

- 1.Auto piederumu interneta veikala prasību definēšana;
- 2.izvēlēto *PHP* ietvaru analīze un izpētīšana;
- 3.labāka un piemērotāka ietvara izvēle;
- 4.veikala izstrādāšana ar izvēlēto ietvaru.

Darba tēma ir aktuāla, jo mūsdienās iespēja iepirkties internetā ir svarīga pircējiem, šāda veida iepirkšanās kļūst vēl aktuālāka. Cilvēku uzticamība internetam aug, šāda veida veikali vairs nav kas jauns un nezināms. Visvairāk cilvēku meklē veikala tīmekļa vietni, gan lai iepazītos ar preču asortimentu, gan lai nopirktu preces izmantojot iespēju netērēt laiku stāvēšanai rindā, vai braukšanai sastrēgumos.

Interneta veikala potenciāls apmeklētājs ir cilvēks, kas vēlas izmantot plašas interneta iespējas arī auto piederumu pirkšanā, vai rezerves daļu pasūtīšanā.

Ta ka interneta veikals reāli tiks izstrādāts, autoresm ir jāizvēlas arī ietvars, kas būs atbilstošs prasībām, saprotams praktiskajā pielietojumā un drošs. Maģistra darba laikā tiks veikta piecu *PHP* ietvaru pētīšana, lai pēc iegūtiem rezultātiem izveidotu auto piederumu interneta veikalu.

Mūsdienās iespējams izmantot visdažādākus līdzekļus, kā izveidot interneta veikalu. Iespējams izmantot gatavas sistēmas, pielāgojot to savām vajadzībām. Piemēram www.yostoro.lv ir maksas sistēma interneta veikaliem, kas ļauj darboties sociālajos tīklos, gan uzturēt savu tīmekļa vietni. Šī sistēma dod iespēju izveidot tieši interneta veikalu ar katalogu, grozu un visu veikalam nepieciešamu. Lai izmantotu šo sistēmu, lietotājam nav vajadzīgas nekādas programēšanas zināšanas. Bet līdz ar to zūd arī iespēja pievienot papildus funkcionālu, tikai

statiskas lapas ar informāciju. Nav lietotāju reģistrācijas, klientiem katru reizi jāievada vienāda informācija par sevi. Bet, kopumā, pircējiem draudzīga sistēma.

Iespējams izmantot arī *CMS*- satura pārvaldības sistēmas. Tas ir programmatūra kura ļauj vienlaikus vairākiem lietotājiem dažādus teksta vai cita veida dokumentus izveidot, apstrādāt un organizēt. Visbiežāk *CMS* tiek saistīts ar internetu. Autori var šo programmu izmantot arī bez jebkādam programmēšanas valodu zināšanām un nereti tieši internetā izveidot jaunus tekstus, papildināt tos ar bildēm, video vai skaņām un publicēt [1]. Piemēram *Joomla*, *Drupal*. Šis variants ir sarežģītāks. Šāda veida sistēmas ētras vienkāršo tīmekļa vietņu izveidošanai, piemēram, vizītkartes tipa vietnei, jeb tīmekļa dienasgrāmatas izveidošanai. Šajā variantā jau var izveidot vairākas izmaiņas un pievienot dažāda veida funkcionālu. Interneta veikala izmantošanai bieži vien jāpieslēdz daudz moduļu, tos savā starpā savienojot.

Viens no variantiem ir izveidot savu sistēmu izmantojot ietvarus. Šāda pieeja ļauj maksimāli ērti izveidot veikalu ar tādu funkcionālu, kas tam ir nepieciešams. Tāda pieeja ir elastīga un droša. Autore izvēlējās tieši šo pieeju, lai izpētītu *PHP* ietvaru iespējas un izveidotu labu, ērtu interneta veikalu, kura iespējams veikt visas nepieciešamas izmaiņas un papildinājumus.

Darbs sastāv no trim daļām. Auto piederumu veikala apraksta daļa paskaidro kāpēc elektroniska komercija ir vērtīga veikaliem, aprakstīta tās klasifikācija un definīcija. Tiks analizētas tās veikala prasības, kas ir nepieciešamas pircējiem un administratoram, lietojot interneta veikalu. Ir izveidots tīmekļa skelets. Otrā daļa ir veltīta ietvaru pētīšanai. Izvēlēti pieci ietvari tiek izpētīti teorētiski un praktiski. Pētīšana notiek pēc vienādiem kritērijiem un vienādos apstākļos. Nodaļas beigās ir izvēlēts ietvars interneta veikala izstrādāšanai. Pēdējā daļā tiek aprakstīta interneta veikala izstrādāšana, tai izvēlētas metodes un paplašinājumi. Galarezultātā ir izstrādāti secinājumi un priekšlikumi.

Darbs kopumā izklāstīts 79 lappusēs, tajā iekļauta 1 tabula un 49 attēli.

2.AUTO PIEDERUMU VEIKALA APRAKSTS

1.1.Interneta veikala priekšrocības.

Interneta veikals ir viena no elektroniskas komercijas sastāvdaļām. Lai saprastu kādas priekšrocības dod šī nozare, autore pievērsīs uzmanību tās definīcijai, aprakstam un veidiem.

Koncepcija par elektronisku komerciju definē to kā *darījumu veikšanu izmantojot informācijas tehnoloģiju un telekomunikāciju līdzekļus*[2]. Šis termins mūsdienās nav nošķirams no interneta un interneta tehnoloģijām.

E-komercijas galvenais mērķis ir paplašināt uzņēmējdarbību, pārceļot to arī virtuālajā vidē, ļaujot pirkt, pārdot, slēgt darījumus un sadarboties, izmantojot modernās tehnoloģijas[3].

Elektroniska komercija aptver vairākas nozares, bet par vienu no populārākiem uzskata tirdzniecību internetā. Interneta dod vairākas priekšrocības, tas padara biznesu par globālu un pieejamu ģeogrāfiski neierobežotos apmēros. E-komercija ir viens no veidiem, kā uzņēmums var sasniegt savus klientus.

Šī nozare ļauj paplašināt uzņēmuma iespējas tirgū, pat nestrādājot ar saviem klientiem reālajā vidē. Tas nozīmē, ka pat ja uzņēmumam nav veikalu tīklu, vai pat veikalu reālajā pasaulē (bricks and mortar), bet ir tikai tīmekļa vietne (clicks and mortar) – tā var būt pilnvērtīga tirdzniecības vieta. Reāla veikala esamība nav obligāts priekšnosacījums sekmīgai e-komercijai.

E-komercijas pamatelementi:

1.E-komercijas subjekti: sistēmas dalībnieki un klienti (biznesa organizācijas, patērētāji).

Auto piederumu e-veikala gadījumā, par e-komercijas subjektiem būs paterētāji, vietnes lietotāji.

2. E-komercijas sistēmas: tirdzniecības, maksājuma, piegādes.

Maģistra darba ietvaros tiks izmantotas piegādes un tirdzniecības sistēmas.

3. E-komercijas procesi:

Tirgus pētīšana,Preču, pakalpojumu un informācijas pārdošana, Pasūtījumu izpildīšana, Darījumu veikšana.

4.E-komercijas tīkli: globālais Internet tīkls, korporatīvie.

Auto piederumu e-veikala gadījumā tiks izmantots globāls internets.

Pēc e-komercijas klasifikācijas, auto piederumu interneta veikals atbilst B2C modelim.

Business to customer- sistēmas ir paredzētas gala patērētājiem, tās ir analogs tradicionālajam veikalam, tikai Internetā (www.2u.lv, www.euroshop.lv).

E-komercijas tiesiskā vide

E-komercijai Latvija tika izstrādāta „Konceptija par elektronisku komerciju” , kas Ministru kabinetā akceptēta 2001.gada 13.martā.

Ta ka Latvija ir Eiropas Savienības (ES) dalībvalsts, tādēļ tiesiskās vides pamatā ir ES regula un direktīva, kas ir iestrādātas Latvijas normatīvajos tiesību aktos.

Pie tam šajā gadījuma jāņem vērā arī patērētāju tiesību likumus, jo pircējas jābūt aizsargātam un drošam, ka viņa nauda tiks godīgi apmainīta pret preci/pakalpojumu, ka arī par to, ka tiks izpildīti visi garantijas, preču maiņas, vai atgriešanas nosacījumi.

Apkopojot visu augšā minēto var secināt, ka elektroniska komercija ir ļoti vērtīga uzņēmumas jo tā:

Paplašina tirgu – ar salīdzinoši mazām investīcijām uzņēmums var sasniegt jaunus klientus, sadarbības partnerus un piegādātājus gan valsts ietvaros gan pasaulē.

Dod iespēju ietaupīt līdzekļus – piemēram, ieviešot E-komercijas risinājumus uzņēmumā samazinās “papīra” informācijas aprites daudzums un tās apkalpošanas izmaksas, kā arī šī aprite notiek efektīvāk.

Uzlabo biznesa organizācijas un procesu efektivitāti – jauni organizācijas modeļi ļauj nodrošināt efektīvāku saikni ar darbiniekiem, ātrāku informācijas apriti, efektīvāku darba organizāciju.

Veicina interaktivitāti - E-komercija veicina uzņēmuma saikni ar klientiem un sadarbības partneriem, ļauj nodrošināt ātru un precīzu atgriezenisko saikni.

1.2.Prasību formulēšana

1.2.1.Apraksts

Topošs auto piederumu interneta veikals dos iespēju potenciālajiem pircējiem iegādāties dažāda veida piederumus automobiļiem. Svarīgi tas, ka šis veikals ir, galvenokārt, domāts vienkāršiem lietotājiem, kuri vēlas nopirkt tieši auto piederumus, kuru iegādei nav jāzina kādi specifiskie termini, jeb kodi, jo ikdienā bez remonta auto apkope ir vajadzīgs daudz citu priekšmetu un šķidrumu.

Šādu preču pirkšanai un pārdošanai nav vajadzīgs tik nopietns katalogs ka auto rezerves daļām, jo šāda veida sistēmai būtu vajadzīgas pavisam citas prasības, kā zināšanas ar šāda veida katalogu strādāšanai. Šāda veida sistēma būtu vajadzīga cita līmeņa darbībai ar citiem klientiem.

Darba mērķis ir izveidot interneta veikalu, papildinot to ar nepieciešamu funkcionalitāti, lai veikals būtu viegs un saprotami struktūrēts. Pamatprasības ir šādas:

- Preču katalogs;
- meklēšanas sistēma;
- informatīva satura lapas;
- preču grozs;
- reģistrācija, autorizācija;
- auto rezerves daļu pasūtīšanas forma.

Preču katalogs

Preču katalogs ir veikala piedāvāto auto piederumu kategoriju saraksts. Dažas kategorijas tiks arī sadalītas apakškategorijās. Saraksta kategorijas, kas attiecas uz auto rezerves daļām, nes sevī informatīva rakstura informāciju, tai skaitā tos izstrādātājus (sk. 1.1. att.), kuru preces ir iespējams pasūtīt interneta veikalā. Šāda veida precēm tiks izmantota speciāla rezerves daļu pasūtīšanas forma. Tas tiks izdarīts šādā veidā, jo veikals ir virzīts galvenokārt uz auto piederumiem, nevis rezerves daļām. Šāda veida precei ir nepieciešams cita veida katalogs,

darbam ar kuru lietotājiem būtu jāzina specifiskas lietas un sīkumus. Izmantojot preču pasūtīšanas formu, pircējam nebūs jāveic saražgīta meklēšana, to izdarīs pārdevējs un piedāvās gatavu risinājumu.



1.1. att. Izgatavotāju saraksta piemērs

Pārējas kategorijas, kas attiecas tieši uz auto piederumiem saturēs preces, kas iespējams pasūtīt caur internetu. Attiecībā no kategorijas būs iespējams redzēt apakškategorijas, kādu sadalījumu pa markām, modeļiem, jeb uzreiz preču saraksts. Tas nozīmē, ka tās preces, kuras neprasa speciāla auto rezerves daļu kataloga izmantošanu, būs izvietotas vietnē līdz ar cenām un aprakstiem, bet pārējas kategorijas (pārsvarā veltītas auto rezerves daļām) nesīs informatīvu raksturu, attēlos tos ražotājus, ar kuriem sadarbojas pārdevējs.

Informatīva satura lapas

Informatīva satura lapas ir svarīgas, lai klientiem būtu pieejama ne vien kontaktinformācija, bet arī tādas lapas kā garantijas, preču piegādes, atgriešanas un pasūtīšanas nosacījumi. Nepieciešamas informācijas saraksts:

- Garantijas nosacījumi;
- piegādes nosacījumi;
- preces atgriešanas nosacījumi;
- kontakti;
- rezerves daļu pasūtīšanas iespējas.

Drošība, ka izvēlēta prece kalpos tik ilgi, cik tas ir paredzēts ražotājam arī ir svarīga. Bieži vien pērkot preces internetā rodas jautājumi par garantiju, remonta, maiņas iespējām un vietām, kur tas ir iespējams. Jebkura prece, kas ir pirktā, šāda veida veikalos pilnīgi atbilst ražotāja garantijai. Līdz ar to arī nosacījumi ir vienādi. Viss, kas ir saistīts par preces atgriešanu/maiņu,

nepieciešamajiem dokumentiem, iemesliem kad to ir iespējams izdarīt būs publicēts attiecīgajā sadaļā.

Mūsdienās cilvēki meklē iespēju patērēt pēc iespējas mazāk naudas par līdzīgiem piedāvājumiem, tāpēc meklē informāciju arī par piegādes iespējām un samaksu par to.

Tīmekļa vietnē arī jābūt informācijas par pakalpojuma sniedzēju, ka arī iespēja ar to sazināties. To varēs izdarīt izmantojot kontaktformu, Skype, jeb elektronisko pastu. Šāda veida informācijas trūkums diezgan bieži rada problēmas, kā arī neticamību pārdevējam. Dažreiz pircējam ir nepieciešams saņemt kāda veida informāciju, viņam pieejamākajā veidā. Daži cilvēki izvēlas sarunu izmantojot telefonu, citi labprāt raksta vēstules. Šāda veida situācijām ir dažādi iemesli, tāpēc ir jācenšas pircējam maksimāli ērti un mierīgi justies.

Preču grozs

Preču grozā pircēji glabās izvēlētas preces, lai vēlāk apstiprinātu pasūtījumu.

Reģistrācija un autorizācija

Reģistrācija

Reģistrācija ir nepieciešama, lai atvieglotu lietotājam turpmāko vietnes lietošanu. Pasūtot preci būs jānorāda dažāda veida kontaktinformāciju un piegādes adresi, lai to nevajadzētu darīt katru reizi var vienreiz reģistrēties. Nākamajā reizē būs jāievada login un parole, visa nepieciešama informācija tiks ņemta no datu bāzes. Šos datus varēs arī nomainīt pasūtīšanas brīdī. Pārdevējs, savukārt, redzēs, ka klients jau ir biežs veikala apmeklētājs, tas dos iespēju uzticēties pircējam, zinot, ka šī nav viņa pirmā iepirkšanās reize. Reģistrācijas brīdī lietotājam tiks piešķirts pircēja statuss, kas ļaus programmai atpazīt piekļuves tiesības autorizācijas brīdī.

Autorizācija

Autorizācija ir svarīga ne vien lietotājiem, bet arī administrātoram. Pēc paroles un lietotājvārda ievadīšanas, programma apstrādās iegūtos datus, apskatīs kādas piekļuves tiesības ir lietotājam. Ja lietotājs ir pircējs viņa iespējas būs tādas, lai veiktu pirkšanas darbības. Administratora tiesības dos iespēju rīkoties ar veikala katalogu, pievienojot, dzēšot, vai labojot informāciju un preces.

Auto rezerves daļu pasūtīšanas iespēja

Šīs tīmekļa vietnes galvenais mērķis ir auto piederumu pārdošana mazumtirdzniecībā, bet tā ka reāli ir iespējams iegādāties arī rezerves daļas, tiek pievienota speciāla detaļu pasūtīšanas forma. Kad pircējs ienāk vienkāršajā jeb „brick-and-mortar” veikalā, atrast pareizo detaļu viņam plīdz pārdevējs, jo šada tipa veikalu un, piemēram, pārtikas veikalu iepirkšanas sarežģītība ir dažāda. Daļa no tiek pircējiem, kas tomēr izvēlas aiziet patstāvīgi uz veikalu, nevis ņemt detaļas autoservisā, tomēr nav profesionāli auto rūpniecībā.

Automobiļu marku un modeļu ir ļoti daudz, kā arī piedāvāto oriģinālo detaļu un to analogu piedāvājums ir ļoti plašs, nav iespējams bez papildus līdzekļiem uzzināt kas tieši ir vajadzīgs. Par palīdzību pārdevējiem kalpo auto rezerves daļu katalogi. Kādreiz tie bija lieli resni žurnāli, kas vairākos eksemplāros glabājas katrā veikalā (šobrīd joprojām tādi tiek pielietoti), pašreiz vairumtirdzniecības detaļu tirgotājiem, kā arī lielajiem veikaliem eksistē speciālas sistēmas datoros, kas atvieglo un paātrina meklēšanas procesu. Šādi katalogi ir pieejami arī lielu tirgotāju klientiem, kas arī ir tirgotāji. Tomēr darbs ar šāda veida katalogu ir diezgan sarežģīts, tāpēc pārdevēji bieži piedāvā pirms darba ar šo sistēmu iziet apmācības kursu. Līdz ar to vienkāršajiem lietotājiem ir vienkārši nevajadzīga šāda sistēma vietnē.

Šāda veida sistēmu gatavie risinājumi ir nopērkami. Veidota interneta veikala pārdevējam ir pieeja šāda veida katalogam. Līdz ar to par vienu no variantiem bija izskatīta forma, kur pircējs ievadītu nepieciešamus datus un pārdevējs varētu apstrādājot pieprasījumu piedāvāt iespējamus rezerves daļu variantus. Taču šī opcija joprojām paliek pat ne par otras kārtas mērķi, bet gan par papildus opciju.

Izanalizējot visu augstāk minēto, autore secināja, ka ir jāizstrādā:

1.Administratīvā daļa (backend):

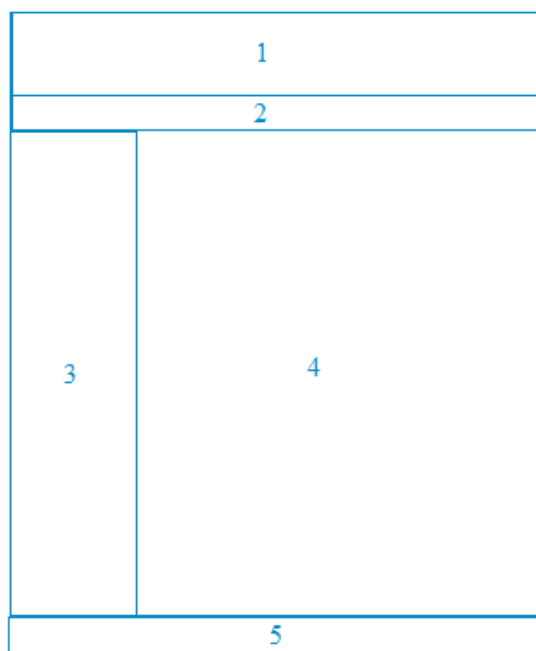
- a. autorizācija;
- b. lietotāju pārvaldība;
- c. kategoriju administrēšana;
- d. preču administrēšana;

2.Lietotāja daļa (frontend):

- a.preču kategoriju navigācija;
- b.tīmekļa galvenā navigācija;
- c.katalogs;
- d.reģistrācija/ autorizācija;
- e.informatīva rakstura lapas;
- f.preču grozs.

1.2.2.Struktūra

Vietnes struktūrai jābūt pēc iespējas vieglākai un intuitīvi saprotamai. Ta sastāv no piecām daļām (sk. 1.2. att.), četras no kurām ir pastāvīgas, nemainīgas, bet viena mainīsies atkarībā no izvēlēta.



1.2. att. Vietnes struktūra

Daļā zem pirmā numura, ir novietots interneta veikala logotips. Otrajā daļā novietota tīmekļa horizontālā navigācija, kas ļaus pircējam pārvietoties pa dažādām lappusēm. Trešajā daļā

ir izvietota vertikāla navigācija, jeb katalogs ar preču kategorijām un apakškategorijām. Bet piektajā daļā atrodas kājene.

Atlikusī ceturtnā daļa ir mainīga. Tās saturs atbilst izvēlētam. Piemēram, izvēloties no kataloga kādu no apakškategorijām, tiks attēlots saraksts ar precēm, kuri tai atbilst. Savukārt, izvēloties sadaļu ar palīdzību, vietnes centrā tiks parādīta informācija lietotājam.

3.IETVARU PĒTĪŠANA

2.1.Pētīšanas kritēriju apraksts

Lai noskaidrotu kāds no izpētītiem ietvariem ir labāks, sākuma ir jāsaprot kādiem nolūkiem tas ir vajadzīgs. Vai vienkārši gribas iemācīties kādu jaunu ietvaru? Vai vajadzīgs ietvars ar lielu moduļu esamību? Varbūt ir vajadzīgs ātrs, adaptējams, jeb drošs ietvars? Jeb svarīga ir dokumentācija, jeb citi avoti, kas veicina ietvara ātru un vieglu apguvi? Tā ka viens ietvars nevarēs atbilst visiem nosacījumiem, sākuma jāatrod svarīgākais sev.

Autores maģistra darba gaita bija jāizstrādā interneta veikals. Tāpēc bija meklēts atbilstīgs ietvars, kas atvieglotu darba procesu un spētu piedāvāt nepieciešamās iespējas. Dokumentācijai bija jābūt pietiekami saprotamai no jauna lietotāja puses.

Ietvaru praktiskās izpētīšanas laikā katrs ietvars bija ienstalēts autores datorā un ar to palīdzību tika veiktas vienādas darbības, lai būtu iespējams salīdzināt iegūtos rezultātus. Ietvari ir analizēti pēc sekojošiem parametriem:

1) Ietvaru apraksts un prasības: iepazīšanās ar ietvaru izstrādātāju dokumentāciju. Kas apraksta ietvara iespējas, svarīgākās īpašības un nepieciešamas darbības ietvara instalēšanai. Informācija tiek ņemta no izstrādātāju oficiālajiem avotiem. Daži ietvaru izstrādātāji piedāvā ne vien pamācības un dokumentāciju interneta vidē, bet arī grāmatas, kas ir pieejamas .pdf formātā.

2) Uzstādīšana: teorētisko instalēšanas darbību pielietošana praksē. Ietvara instalācija autores datorā ar mērķi, tā tālākai pētīšanai. Tika novērtēts instalēšanas sarežģītums un dokumentācijā aprakstītā vērtīgums.

3) Programmu izstrādāšana: pēc kārtas tika ieinstalēti un izpētīti visi, darbā aprakstīti ietvari. Ar katra ietvara palīdzību bija izstrādātas divas vienādas programmas. Pirmā, „Hello world” programma, kurā izmantojot MVC arhitektūru, ar kontrolleri un skatu, uz ekrāna tika izvadīts teksts “Hello world!” ar lapas virsrakstu “My title”.

Noformējums un teksts bija vienāds visiem ietvariem. Otrā programma parādīja darbu ar datu bāzi. Tika sagatavota viena datu bāze “testdb” ar tabulu “Testtabs”.

Tabula saturēja divas kolonnas „Id” un „col1”, ar datiem (sk. 2.1. att.).

| Id | col1 |
|----|------|
| 1 | 10 |
| 2 | 5 |
| 3 | 1 |

2.1. att. Tabulas „Testtabs” saturs

Programmai, izmantojot *MVC* arhitektūru, bija jāizvada informācija no datu bāzes. Arī šajā piemērā noformējums un informācija bija vienāda visiem ietvariem. Rezultāti ir attēloti 2.2 attēlā.

| | |
|---|----|
| 1 | 10 |
| 2 | 5 |
| 3 | 1 |

2.2. att. Programmas izvadadīta rezultāta piemērs

4) Ātrums: Testēšanai tika izmantots *ApacheBench* (*AB*), kas ietilpst standarta *Apache* pakotnē. Tās būtība ir noteikta skaita http-pieprasījumu sūtīšanā uz servera, saņemot pretī to apstrādāšanai izmantota laika mērījumu rezultātus. Visi ietvari bija testēti uz viena datora, ar vienu rīku. Ātrums tika mērīts abām izstrādātām programmām. Katrs ietvars tika ieinstalēts attiecīgi izstrādātāju pamata prasībām, nepievienojot nekādus papildus modulus, vai paātrinātājus [4].

Rezultāti ir attēloti pieprasījumos sekundē (Requests Per Second, *RPS*), tāpēc, lielākas vērtības ir labākas.

AB programma piedāvā vairākus veidus, kā var novērot slodzi tīmekļa vietnē. Iespējams norādīt dažādus parametrus (pēdēja *AB* versija uztur 25 parametrus), piemēram, -t nozīmē, atļauto sekunžu maksimumu, patērēto pieprasījumu apstrādāšanai. Lai veiktu detalizētāku atskaiti, janorāda -c paramerts, kas nosaka vienlaicīgi sūtīto pieprasījumu skaitu. Tas nozīmēs, ka pieprasījumi tiks sūtīti pa porcijām, kas palielina slodzi. Piemēram, testējot slodzi ar šādiem

kritērijiem (ab -t 30 -c 10 http://localhost/index.php), *AB* parādīs cik pieprasījumu sekundē tiks apstrādājot sūtot pa 10 pieprasījumiem 30 sekunžu laika.

Autore piedāvā apskatīt paša iegūtus rezultātus, kā arī iepazīties ar Yii komandas izstrādātāju veiktajiem testiem.

5) Drošība: tika apskatīti ietvara izstrādātāju piedāvātie drošības pasākumi, kas ir aktuāli projektam;

6) Dokumentācija: tika izpētīta izstrādātāju dokumentācija, kas domāta ietvara iepazīšanai un pirmo soļu (programmu izstrādāšanā) veikšanai. Tā ka ietvari tika lietoti pirmo reizi, tāpēc dokumentācijas pilnīgums bija vērtēts no jauna lietotāja viedokļa. Cik dokumentācija spēj palīdzēt izprast ietvara būtību un to loģiku.

7) Citi: ja ietvars piedāvāja kādu interesantu risinājumu, rīku, moduļu, iespēju, kas ir vērtīgs projekta izstrādāšanai, tas tika aprakstīts šajā nodaļā.

Lai veiktu pētīšanu uz autores datora tika ieinstalēti šādi rīki: *Apache 2.2* serveris, *ApacheBench*, *PHP 5.3.5*, *MySQL Server 5.1*, *Notepad++*.

2.2.Ietvaru pētījuma rezultātu apraksts

2.2.1.Ietvaru apraksts un prasības

CodeIgniter

CodeIgniter — Populārs *MVC* ietvars ar atvērtu kodu, pilnvērtīgu tīmekļa sistēmu izstrādāšanai. Ietvars atšķiras ar vienkāršību, kura tiek sasniegta pateicoties sekojošiem faktoriem:

- Pilna un kvalitatīva dokumentācija ar piemēriem, ka arī *Wiki*;
- liels daudzums ar palīgmateriāliem- apmācībām, kas sasniedzami oficiālajā vietnē, ka arī vairākos citos resursos;
- programētājam nav jāmācās lietot koda ģenerātoru no komandrindas;
- CodeIgniter* ir viens no ātrākiem un resursu neprasošiem ietvariem;
- sākuma pakas izmērs ir ļoti mazs (2.02. versijas izmērs ir 2.2 mb) [5].

Prasības:

3.5.1.6. versijas *PHP* (vai jaunāks).

4. Vairākumam lietotnēm, datu bāze nav vajadzīga. Pēdēja versija strādār ar: *MySQL (4.1+)*, *MySQLi*, *MS SQL*, *Postgre*, *Oracle*, *SQLite un ODBC*. [6]

CodeInteger piedāvā:

- Strādā ar dažādām datu bāzēm, piemēram: *MySQL*, *PostgreSQL*, *MSSQL*, *SQLite*, *Oracle*;
- *CodeIgniter* saprot pseido-*ActiveRecord*, kas līdzīga ar SQL sintaksi;
- sistēma viegli papildināma izmantojot pašrakstītas, vai piedāvātas bibliotēkas, kuras var parveidot. Ietvars satur vairākas bibliotēkas, kas nepieciešamas darbam ar failiem, e-pastu sūtīšanai, formu validācijai, sesiju uzturēšanai, darbam ar attēliem un citiem;
- eksistē pakešu menedžeris *Sparks*, kas palīdz viegli uzstādīt ārējas bibliotēkas no komandrindas;
- *PHP ActiveRecord*, *Doctrine*, *Propel* un daži citi;
- iespējams izmantot *Smarty*, *TinyButStrong* (šablonizatori) [5].

Zend Framework

Zend framework – ir *PHP* ietvars, kas izstrādāts ar *Zend* kompānijas strādniekiem, kuri ir tieši *PHP* valodas autori. Tāpēc tas turpina *PHP* tradīcijas, bāzējas uz saprotamības, vienkāršības, objekt-orientētiem principiem [7].

Zend Framework ir drošs, viegli uztverams un moderns. Ietvars pamatots uz *MVC* idejas. *Zend framework* satur *MVC* komponentus, vairākas bibliotēkas, kas ir vērtīgas lietotņu realizēšanai, eksistē komponentes integrācijai ar dažādiem servisiem, ka *YouTube*. Dažas iespējas, ko piedāvā *Zend framework*: [8].

- Visi komponenti ir objekt-orientēti ar *PHP 5* savienojami;
- viegli uztverama arhitektūra;
- *MVC*;

- uztur dažādas datubāzes: MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server;
- e-pastu veidošana un sūtīšana ar *mbox*, *Maildir*, *POP3* un *IMAP4*;
- vienībtesti.

Prasības:

- *PHP 5.2.4* (vai augstāk)
- *Web servers* ar *mod_rewrite* funkciju[9].

Symphony

Symfony ir *MVC* ietvars ar atvērtu kodu. To izstrādātāji Sensio Labs aģentūra, sākuma to izstrādāja savu klientu tīmekļa vietņu izveidošanai. Bet 2005. Gadā, tas tika publicēts „Open Source” licenci. Symfony izstrādātāji piedāvā plašu dokumentāciju un profesionālu atbalstu [10].

Symfony piedāvā:

- Viegli instalējams un konfigurējams ietvats;
- neatkarīgs no datu bāzes;
- viegli lietojams, lokans, pielietojams sarežģītu uzdevumu risināšanai;
- pielietojami dizaina šabloni;
- stabils ilgtermiņa projektiem;
- viegli lasāms kods ar komentāriem ;
- viegli paplašināms ar citu izstrādātājie bibliotēkām [11]

Prasības:

- *PHP 5.3.2*
- iespējots *Sqlite3*
- iespējots *JSON*
- iespējots *ctype*
- *PHP.ini* ar noteikumiem laika zonas (*date.timezone*) nostādījumiem [12]

CakePHP

CakePHP ir bezmaksas un brīvs kods, kas ārti attīstās. Tās pamatā ir struktūra, kas domāta tīmekļa veidošanai. *CakePHP* piedāvā visus kodēšanai nepieciešamus instrumentus. Viss kas jāizdara programētājam, ir jāizdomā sava tīmekļa specifiska loģika. Ietvars ir domāts, lai atkārtotu procesu izdomāšanas vietā, izmantotu gatavu risinājumu, nokonfigurētu atbilstoši prasībām.

Dažas iespējas, ko piedāvā *CakePHP*:

- Aktīvs projekts, kas pastāvīgi uzlabojas un atjaunojas;
- bezmaksas, atvērts kods;
- draudzīgs ar *PHP* 4. un 5. versiju;
- integrēts darbam ar datu bāzēm;
- koda ģenerēšana;
- MVC* arhitektūra;
- dispetčers, kas pārveido *URL*;
- iebūvēta datu validācija;
- šablonu sistēma ir ātra un ērta;
- palīgi *AJAX*, *JavaScript*, *HTML*-formu attēlošanai;
- komponentes darbam ar Email, Cookie, drošību, sesijām un pieprasījumu apstrādāšanu
- automātiska datu apstrādāšana;

Prasības

- HTTP* Server. *Apache* ar *mod_rewrite* (ir labi, bet nav obligāti);
- PHP* 4.3.2 vai augstāk;
- Tehniski datu bāze nav obligāta, bet vairākums no aplikācijām bez tās nevarēs strādāt.
- CakePHP* strādā ar šādām datu bāzēm:
 - a.*MySQL* (4 vai augstāk);
 - b.*PostgreSQL*;
 - c.*Firebird DB2*;
 - d.*Microsoft SQL Server*;

- e.Oracle;
- f.SQLite;
- g.ODBC;
- h.ADOdb [13] .

Yii Framework

Yii — ir augstieffektīvs komponentpamatots *PHP* ietvars, kas domāts liela mēroga tīmekļa projektu izveidošanai. Tas ļauj maksimāli izmantot atkārtota koda izmantošanas koncepciju un paātrināt izstrādāšanas procesu. Nosaukums nozīmē *Yi* [ji:], kas nozīmē vienkāršs (*easy*), efektīvs (*efficient*) un paplašināms (*extensible*) [14].

Diezgan maz informācijas par to, ko tieši piedāva *Yii*. Tā ir atrodamā pa tiešo dokumentācijā.

Pēc tās analīzes kļuva skaidrs, kas *Yii* arī piedāva:

- MVC* struktūra;
- DAO* un *ActiveRecord* interfeisi darbam ar datu bāzēm;
- formu validācija;
- autentifikācija un autorizācija;
- AJAX* izmantošana, integrēšana ar *jQuery*;
- kļūdu novērošana un žurnālēšana;
- drošība;
- bāzes *PHP* ģenerēšana priekš *CRUD* operācijām;
- tēmu noformējums, to viegla maiņa;
- iespēja pieslēgt ārējas bibliotēkas;
- ļoti labi izstrādāta dokumentācija;
- automātiska testēšana [15].

Prasības

Darbam ar *Yii* ir nepieciešams *PHP 5.1* (tas ir minimums). Der *Apache HTTP* server zem *Windows/Linux*, vai citām platformām, kas uztur *PHP 5.1*. [16]

Pēc *Yii* uzstādīšanas jāparbauda, vai serveris atbilst prasībām, tas izdarāms uzejot pēc saites <http://hostname/path/to/yii/requirements/index.php> .

Secinājumi

Kopumā apskatot katra piedāvājuma sistēmas prasības, kļūva skaidrs, ka projektā var tikt lietots jebkurš ietvars, jo ir izmantots *HTTP Server Apache* ar *mod_rewrite* , 5.2.17 versijas *PHP* un *MySQL* datu bāze. Visi ietvari tiek būvēti uz objekt-orientētas struktūras un izmanto MVC kontrollera arhitektūru. Tā ka šis kritērijs bija teorētiskā, nevis praktiskā puse un tas neparāda, cik praksē ērti izmantot kādu no ietvariem. Tāpēc kopējā vērtējuma tabulā par šo punkti netiks likti. *CodeIgniter* liekas ļoti pievilcīgs, jo iepazīšanās dokumentācija ir izstrādāta maksimāli saprotami. Lai atrastu *Yii* iespējas cītīgi jāapskata dokumentācija, bet galā ietvars liekas ļoti spēcīgs. *Zend Framework* izstrādātāji piedāvā ļoti daudz informācijas, kura ir grūti atrast nepieciešamo. Pat ietvara apraksts, prasības un iespējas ir grūti atrodamas un saliekamas kopā no dažādām lapām.

2.2.2. Uzstādīšana

CodeIgniter

Ļoti viegli atrast instalācijas saiti. Uzstādīšana ir triviāla, pēc arhīva lejupielādēšanas, faili ir jāizpako un jāieliek izvēlētajā direktorijā. Faili ir gatavi darbam. Pēc noklusējuma failam *index.php* ir jāatrodas saknes direktorijā. Failā *application/config/config.php* jāpieraksta un jānostāda saknes url (*\$config['base_url'] = 'http://localhost/;*). Darbam ar datu bāzi jāaver un jāieraksta nostādījumi failā *application/config/database*.

```
$db['default']['hostname'] = 'localhost';
```

```
$db['default']['username'] = 'root';
```

```
$db['default']['password'] = 'root';
```

```
$db['default']['database'] = 'testdb';
```

Failu konfigurācija nesastādīja nekādas grūtības, viss pietiekami skaidrs un ātri izpildāms.

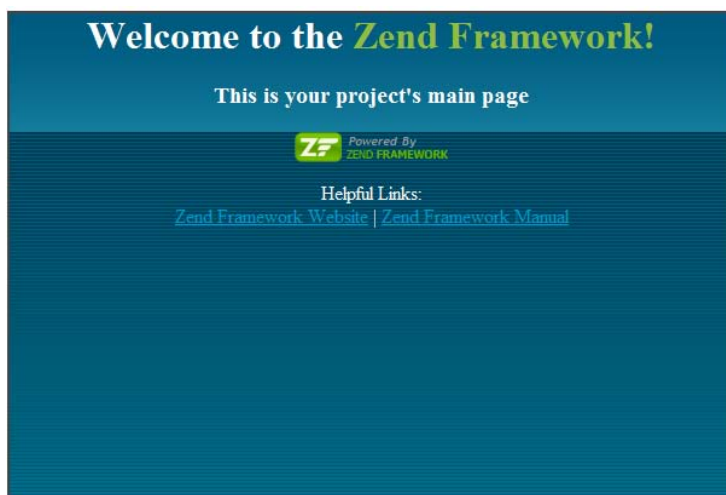
Uzstādīšana viegla, nerodas nekādas problēmas, kuras jārisina ar citu avotu palīdzību. Viss strādā ar pirmo reizi.

Ietvara instalēšanai tika lietota instrukcija izstrādātāju oficiālajā vietnē [16].

Zend Framework

Lai lejupielādētu *Zend Framework* arhīvs ir nepieciešama reģistrācija, kas padara procesu ilgāku. Pēc izstrādātāju pamācības, instalācija notiek viegli, taču lai viss strādātu autoresm bija jāmeklē papildus pamācības, kā tikt galā ar šo ietvaru. Rezultātā, vērtīgākais veids, kā sākt dabu ar projektu, bij atrodams habrahabr.ru interneta resursā [17].

Pēc arhīva ielādēšanas, saknes direktoriņā bija jāiekopē mape ar bibliotēku, tad jāuzkonfigurē mainīgas vides (path). Tas bija nepieciešams, lai ar komandrindu varētu izveidot jaunu projektu, ar *Zend* failu struktūru. Beigās, kad projekts tika izveidots, ziņojums, ka viss ir izveidots veiksmīgi (sk. 2.3. att.), nerādījās, ta vieta autoresm ziņoja par kļūdu. Kaut kan visas darbības tika izdarītas pēc instrukcijas, bibliotēku bija jāiekopē jaunajā projektā patstāvīgi.



2.3. att. **Zend** ietvara ziņojums

Symphony

Lai sāktu darbu ar *Symfony*, tika izvēlēta pēdēja stabila ietvara versija 2.0. Ietvara instalēšanai tika lietota instrukcija izstrādātāju grāmatā *Symfony Book 2.0* pdf formatā [18].

Sākuma ir jāielādē un jāatrhivē piedāvatais arhīvs ar *Symfony 2* servera saknes direktoriņā. Pēc tam, palaižot `config.php` failu, programma parāda, kādus pielikumus jāpievieno serverim, vai

to, ka viss ir kārtībā. *Symfony* piedāvā veikt datu bāzes uzstādījumus izmantojot saiti, kas atvērot izskatās kā attēlā 2.4.

STEP 1 > STEP 2

Configure your Database

If your website needs a database connection, please configure it here.

| | |
|-------------|----------------|
| Driver * | User * |
| MySQL (PDO) | root |
| Host * | Password |
| localhost | |
| Name * | Password again |
| testdb | |
| Port | |
| | |




NEXT STEP

2.4. att. Symfony datu bāzes uzkonfigurēšana

Iespējams arī veikt izmaiņas failā `app/config/parameters.ini`. Failā, jeb pēc piedāvātas url adreses jānorada datu bāzes konfigurācijas- parole, lietotājvārds, nosaukums, utt. Nākamajā solī tiek parādīts sveicināšanas ekrāns (sk. 2.5. att.)

Welcome!

Congratulations! You have successfully installed a new Symfony application.

| | | |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  READ THE QUICK TOUR |  CONFIGURE |  RUN THE DEMO |
| Documentation The book The cookbook Glossary | Contributing Trainings Books | Community IRC channel Mailing lists Forum How to be involved Contributing |

2.5. att. Symfony sveicināšanas ekrāns

Konfigurējot *Symfony*, izmantojot .tgz arhīvu, bija konstatēta kļūda, kuras dēļ, automātiskā datu bāzes konfigurēšana nestrādāja. Izmantojot .zip formātā arhivētus failus viss strādāja ar pirmo reizi.

Ietvars veiksmīgi tiek uzkonfigurēts ar pirmo reizi bez jebkādām problēmām.

Lai paaugstinātu ietvara ātrumu izstrādātāji iesaka izmantot kādu no *PHP* paātrinātājiem, piemēram, *APC*. Tā ka iepriekšējos ietvaros šāda iespēja nebija izmantota un ātrums tika mērīts bez efektivitātes uzlabošanas, arī šajā gadījumā, nekādi papildus pielikumi nebija pievienoti. Ar to var paskaidrot diezgan zemu efektivitāti, tās mērīšanas rezultātā ar *Apache* bench.

CakePHP

Sākumā tika ielādēta CakePHP 2.0.6 Stable versija. Pēc ietvara vārdiem, šī ir stabila versija, pašreiz eksistē arī 2.1 beta versija. Uzstādīšana veicama diezgan viegli pēc izstrādātāji norādījumiem oficiālajā tīmekļa vietnē [13]. Jāielādē arhīvs un jāuzliek faili servera saknes direktoriijā. Bet CakePHP izstrādātāji rekomendē mainīt arī Apache uzstādījumus, lai DocumentRoot ceļš būtu "D:/site/localhost/www/app/webroot". Bet tas nav obligāti, tikai ērtībai.

Pēc uzstādīšanas tiek parādīta kļūda header rajonā, tā ziņo, par kļūdu laika joslas uzstādīšanā. Izmaiņas jāveic PHP konfigurācijas failā, kļūda neattiecas uz ietvaru. Liekas, ka neskatoties uz kļūdu header daļā, tomēr programma ir uzstādīta, tāpēc arī kopējā vērtējumā tā neko nenožīmēs.

Par nākamo uzstādīšanas soli ir datu bāzes pievienošana. Izmaiņas jāveic failā app/config/database.php. Tika pievienoti nepieciešami dati.

```
public $default = array(  
    'datasource' => 'Database/Mysql',  
    'persistent' => false,  
    'host' => 'localhost',  
    'login' => 'root',
```

```
'password' => 'root',  
'database' => 'testdb',  
'prefix' => "",  
'encoding' => 'utf8',);
```

Pēc visu nepieciešamu soļu izpildīšanas un atvērot `http://localhost`, uz ekrāna parādījās tabula ar darba rezultātiem. Uzraksti nozīmēja, ka uzstādīšanas rezultāta tika pamanītas, ka 3 no 5 soļiem ir izpildīti pareizi. 1 kļūda bija labojama izmainot faila `app/config/core.php` rindiņu „`Configure::write('Security.salt', 'DYhG93b0qyJfIxfS2guVoUubWwvniR2G0FgaC9mi');`” uz „`Configure::write('Security.salt', 'DYhG93b0qyJfIxfS2guVoUubWwvniR2G0FgaC9Ni');`”. Tas nozīmē, ka bij jāmaina hash vai nu pilnība, vai tikai dažus simbolus.

Otrā kļūda bija līdzīgi novērsta tajā pašā failā, rindiņa „`Configure::write('Security.cipherSeed', '76859309657453542496749683645');`”, kļūva par „`Configure::write('Security.cipherSeed', '76859309657453542496749683644');`”

- Notice (1024): Please change the value of 'Security.salt' in `app/config/core.php` to a salt value specific to your application [`CORE\cake\libs\debugger.php`, line 697]
- Notice (1024): Please change the value of 'Security.cipherSeed' in `app/config/core.php` to a numeric (digits only) seed value specific to your application [`CORE\cake\libs\debugger.php`, line 701]
- Your tmp directory is writable.
- The FileEngine is being used for caching. To change the config edit `APP/config/core.php`
- Your database configuration file is present.
- Cake is NOT able to connect to the database.

Abi šie risinājumi tika aprakstīti *CakePHP* dokumentācijā komentāros.

Tālāk viss gāja labi, bet pēdējais punkts nozīmē, ka *CakePHP* nevar pieslēgties norādītai datubāzei. Ta ka ievadīti dati (`'host' => 'localhost', 'login' => 'root', 'password' => 'root', 'database' => 'testdb'`;) bija pareizi un 100% strādājoši, autoreism bija jāvelta daudz laika šīs problēmas risināšanai. Visi ieteikumi, pieejami dokumentācijā un citu lietotāju komentāros bija apmēram par datu pareizības pārbaudi un datu bāzes darbaspēju. Tā ka šie dati un datu bāze jau tika

notestēti iepriekšējam ietvaram un viss strādāja korekti, šajā ziņā kļūdu vairs nevarēja būt. Pēc CakePHP atkārtotas instalēšanas rezultāti bija vienādi. Ietvars negribēja pieslēgties datu bāzei.

Problēma tika atrisināta ielādējot un ieinstalējot *CakePHP* vecāku versiju 1.3. Šaj versijai datu bāzes uzstādījumi bija korekti, neskatoties uz to, ka tie nemaz neatšķirās no iepriekšējiem. Galā tika sasniegtas gaidītas veiksmīgas *CakePHP* instalēšanas rindiņas (sk. 2.6. att.). Darbam tika izmantotas pamācības no CakePHP grāmatas [19].

```
Your tmp directory is writable.
```

```
The FileEngine is being used for caching. To change the config edit APP/config/core.php
```

```
Your database configuration file is present.
```

```
Cake is able to connect to the database.
```

2.6. att. CakePHP veiksmīgas uzstādīšanas rezultāts

Tā ka ietvara uzstādīšana netika veikta ar pirmo reizi, pat vispār nesanāca pēdējā stabilā versijā, bet gan veiksmīgi un ātri paveikta vecākai versijai, ietvara uzstādīšana tiek vērtēta ar 3.

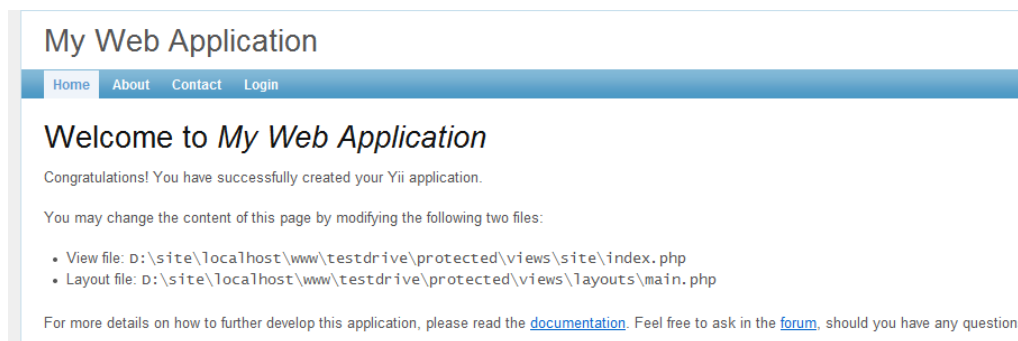
Yii Framework

Uzstādīšana ļoti vienkārša (tiek veikta izmantojot Yii pamācību 1.1 versijai [16]). Sākumā jāizpako Yii arhīvs saknes direktorijā. Tad komandrindā ieraksta sekojošas komandas:

```
„D:/site/localhost/www/yii/framework/yiic webapp D:/site/localhost/www/testdrive”
```

```
„Create a Web application under 'D:\site\localhost\www\testdrive'? [yes|no] yes”
```

Pēc šīm darbībām izveidojas jauns projekts. Atvērot to pārlūkprogrammā, parādās ziņa, ka programma ir izveidota (sk. 2.7. att.).



2.7. att. Yii Framework veiksmīgas uzstādīšanas ekrāns

Testa programmā, izstrādātāji uzreiz piedāvā lietotājam apskatīt ietvara iespējas. To var izdarīt pastrādājot ar navigācijas pogām „Home”, „About”, „Contact”, „Login”.

Tas ir diezgan vērtīgi, jo uzreiz bez papildus darbībām, ir iespējams apskatīties, kā izskatās formas un ko tās prot darīt.

Lai pieslēgtos datu bāzei ir jāveic korekcijas failā testdrive/protected/config/main.php.

```
'db'=>array(  
  
    'connectionString' => 'mysql:host=localhost;dbname=testdb',  
  
    'emulatePrepare' => true,  
  
    'username' => 'root',  
  
    'password' => 'root',  
  
    'charset' => 'utf8',)
```

Datu bāze ir pievienota.

Visas darbības izpildāmas viegli, ātri un saprotami, viss strādā uzreiz.

Secinājumi.

CodeIgniter uzstādīšana bija viegla, neradās nekādas problēmas, kuras jārisina ar citu avotu palīdzību. Viss strādā ar pirmo reizi.

Pirmie soļi ar *Zend Framework* bija diezgan sarežģīti un nesaprotami, lai veiksmīgi tiktu galā, bija jāmeklē papildus resursi ar palīdzību. Šādi resursi ir daudz un tie piedāvā dažādus instalācijas variantus. Tomēr veiksmīgi ieinstalējot *Zend Framework*, otrajā reizē jautājumu vairāk nebūs. Līdz ar to, autore secina, ka *Zend* lietošana ir virzīta vairāk uz pieredzējušajiem lietotājiem, savukārt jauniem lietotājiem, tā sagādā grūtības un rada jautājumus.

Strādājot ar *Symfony*, nebija konstatētas īpašas grūtības, izņemot to, ka izmantojot .tgz arhīvu, bija konstatēta kļūda, kuras dēļ, automātiskā datu bāzes konfigurēšana nestrādāja. Izmantojot .zip formātā arhivētus failus viss strādāja ar pirmo reizi. Šīs kļūdas risinājums bija atrasts dokumentācijā, tāpēc tās labošana nesagādāja grūtības.

CakePHP uzstādīšana netika paveikta izmantojot pēdējo stabilu versiju 2.0.6. Risinājums tā arī netika atrasts. Tomēr vecāka ietvara versija 1.3 tika veiksmīgi un ātri uzinstalēta. Istenībā *CakePHP* prasa veikt vairākas korekcijas, lai sasniegtu mērķi. Tā, ka atšķirība no situācijas ar *Symfony*, (kur arī bija jāielādē arhīvs atkārtoti), kļūdas novēršana nebija aprakstīta dokumentācijā, kā arī nekur citur, ietvara uzstādīšana tiek vērtēta ar 3.

Yii ietvars tika uzstādīts bez jebkādam problēmām, viss bija skaidri aprakstīts dokumentācijā.

Par uzstādīšanas „līderiem”, pēc autores domām, ir *CodeIgniter* un *Yii framework*.

2.2.3. Programmu izstrādāšana

CodeIgniter

Programmu izstrādāšanai tika lietota 2.1.0 versijas dokumentācija [6], kas ir atbilstoša ietvara versijai. Ar ietvaru versiju maiņu dokumentācija arī mainījās, līdz ar to vecās dokumentācijas izmantošana radīja kļūdas „Hello world!” programmas izstrādāšanā, jo galvenā kontrolera nosaukums mainījās līdz ar versijām. Nesaskaņas ātri tika novērstas atverot pareizo dokumentācijas versiju.

Tā ka *CodeInteger* ietvars ir pamatots uz *MVC*, tā failu struktūra atbilst prasībām. Mapē *application/Controllers* atrodas kontrolleri, taču *application/view* glabājas skati.

Sākumā tika izveidots kontrolleris *Hello.php* (sk. 2.8. att.). Pēc tam tika izveidots skats *Helloview.php* (sk.2.9. att.).

```
w.php hello.php
<?php
class Hello extends CI_Controller {

    function index()
    {
        $data['title'] = "My Real Title";
        $data['hello'] = "Hello world!";

        $this->load->view('helloview', $data);
    }
}
```

2.8. att. *Hello.php* kontrolleris

```
helloview.php hello.php
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 |
5 <html>
6 <head>
7 <title><?php echo $title;?></title>
8 </head>
9 <body>
10 <h1><?php echo $hello;?></h1>
11 </body>
12 </html>
13
```

2.9. att. *Helloview.php* skats

Programma strādāja ar pirmo mēģinājumu. Dokumentācija ļoti skaidri un īsi paskaidro visu galveno, parāda ar piemēriem. *CodeIgniter* dokumentācija atstāja labu iespaidu. Neradās ne jautājumu, ne grūtību. Viss ļoti skaidri un saprotami paskaidrots.

Darbs ar datu bāzi

Lai strādātu ar datu bāzi, sākumā pie tās ir jāpieslēdzas. *CodeIgniter* piedāvā divus variantus: automātiskā, vai manuāla pieslēgšanās. Automātiskā pieslēgšanās, nozīmē, ka opcija "auto connect" lādēs klasi DB vienmēr, katrai lapai. Lai to izdarītu jāpievieno vārds „database” masīvam \$library failā application/config/autoload.php.gadījumā, ja pieslēgšanās katrā tīmekļa vietnes lappusē nav obligāta (autores vietnei tas nav vajadzīgs), ir iespējams pieslēgties katrai lapai atsevišķi. Lai to sasniegtu, jāpievieno funkcija \$this->load->database(); vajadzīgam kontrollerim.

Lai pamēģinātu darbu ar datu bāzi tika izveidoti divi faili:

Kontrolleris db.php (sk. 2.10. att.)

```
]<?php
class Db extends CI_Controller {

    function index()
    {
        $this->load->database();
        $this->load->view('dbview');
    }
}
-?>
```

2.10. att. db.php kontrolleris

Un skats dbview.php (sk. 2.11 att.)

```

<html>
  <head>
    <title><?php echo "Connect to DB";?></title>
  </head>
  <body>
    <table>
      <?php
        $query = $this->db->query('SELECT Id, col1 FROM test_tab');
        foreach ($query->result_array() as $row)
        {
          echo "<tr>";
          echo "<td>".$row['Id']."</td>";
          echo "<td>".$row['col1']."</td>";
          echo "</tr>";
        }
      ?>
    </table>
  </body>
</html>

```

2.11. att. Dbview.php skats

Zend Framework

Arī *Zend Framework* piedāvā savu struktūru failu glabāšanai. Tā ir diezgan vienkārša un saprotama. Kontrolleri glabājas projekta mapē `application\controllers`. Kā arī pārējos ietvaros kontrolera nosaukums atbilst adresei, pēc kuras var tikt līdz skatam. Piemēram, `CnameController`, atbildīs par skatu `cname`. Kas būs sasniedzams pēc url `http://localhost/cname`.

Programmu izstrādāšanai tika lietota *Zend Framework* izstrādātāju dokumentācija [24]. „Hello world!” programmas izveidošanai tika izveidots kontroleris, kuru var apskatīt zemāk (sk.2.12. att.).

```
IndexController.php

<?php

class IndexController extends Zend_Controller_Action
{

    public function indexAction()
    {
        $this->view->title = "My title";
        $this->view->text = "Hello world!";
    }
}
```

2.12. att. IndexController.php kontroleris

Kontrolerī jādefinē nosaukums un mainīgais ar tā saturu. Tas viss sūtas uz attiecīgu skatu (sk. 2.13. att.), kas glabājas views\scripts\index mapē. Kur index ir kontrolera pirmā daļa- IndexController, mapes ar skatu nosaukums un šablona nosaukums. Šabloni tiek glabāti .phtml formatā.

```
index.phtml

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <title><?php echo $this->title;?></title>
6 </head>
7 <body>
8 <h1><?php echo $this->text;?></h1>
9 </body>
10 </html>
11
```

2.13. att. index.phtml skats

Šīs programmas izstrādāšana bija diezgan viegla, kaut gan skaidrojums bija jāmeklē citos resursos, jo *Zend Framework* izstrādātāju vietnē, informācija šķiet, ir veltīta „pieredzējušākiem,, ietvara lietotājiem.

Darbs ar datu bāzi

Lai sāktu darbu ar datu bāzi, jānokonfigurē fails `application.ini`, kur jānorāda informāciju, par piekļuvi datu bāzē:

```
[development : production]

phpSettings.display_startup_errors = 1

phpSettings.display_errors = 1

resources.frontController.params.displayExceptions = 1

resources.db.adapter = "PDO_MYSQL"

resources.db.params.host = "127.0.0.1"

resources.db.params.username = "root"

resources.db.params.password = "root"

resources.db.params.dbname = "testdb"
```

pēdējas 5 rindas ir jāpievieno ar saviem piekļuves datiem. Tālāk ir jāpieslēdz `PDO_MYSQL` adapteri, darbam ar datu bāzi.

Ar komandrindā ievadītu komandu `php create db-table Testtabs Testatabs`, automātiski tiks izveidota mape `application/models/DbTable` un fails `Testtabs.php`. `testtabs` ir tabulas nosaukums. Jaunizveidotajā modelī būs ierakstīta klase ar tabulas nosaukumu. Failu ir iespējams papildināt ar metodēm, kas aprakstīs operācijas, kuras būs nepieciešamas darbam. Testa piemēra izveidošanai, pietiek ar tabulas nosaukuma definēšanu: `protected $_name = 'testtabs';`. Modeļa nosaukums ir ļoti garš. Šāda veida struktūra ir nepieciešama programmai, lai ietvars varētu

noteikt, kāds fails ir jāpieslēdz, kad tiks izveidots šī modeļa objekts. Šis nosaukums saka, ka modelis atrodas katalogā application/models/db-table, bet pats modelis nosaucas Testtabs.

(sk. 2.14. att.)

```
Controller.php | Testtabs.php | index.phtml
<?php
class Application_Model_DbTable_Testtabs extends Zend_Db_Table_Abstract
{
    protected $_name = 'Testtabs';
}
```

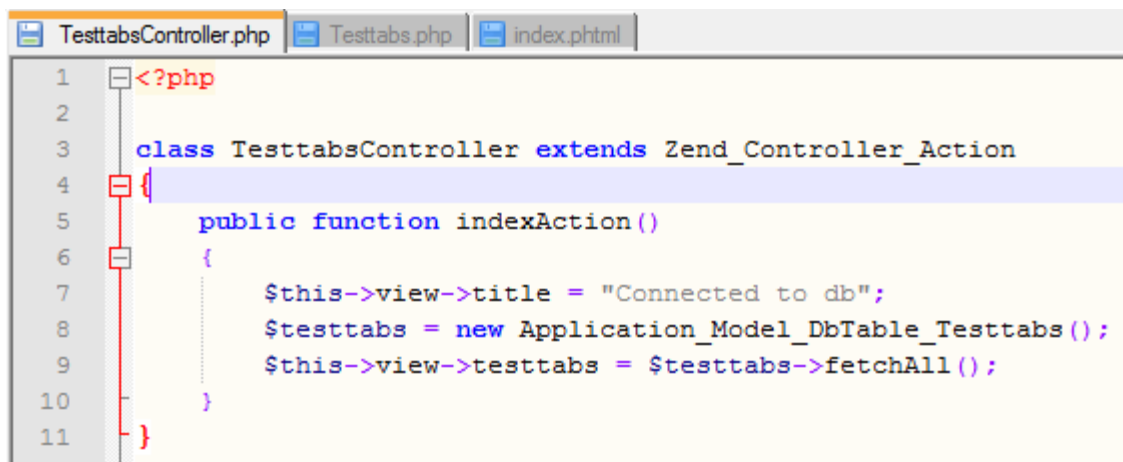
2.14. att. Testtabs.php modelis

Nākamais solis ir šablona izveidošana, kas arī ir skats (sk. 2.15. att.).

```
Controller.php | Testtabs.php | index.phtml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title><?php echo $this->title;?></title>
  </head>
  <body>
    <table>
      <?php foreach($this->testtabs as $testtab) : ?>
      <tr>
        <td><?php echo $testtab->Id;?></td>
        <td><?php echo $testtab->col1;?></td>
      </tr>
      <?php endforeach; ?>
    </table>
  </body>
</html>
```

2.15. att. index.phtml skats

Kā kļūst redzams, iepazīstoties ar kontrolleri (sk. 2.16. att.), lai izvadītu uz ekrāna kolonnas no tabulas, jādefinē vajadzīgais modelis un ar komandu `fetchAll()`, skatam kļūs pieejams masīvs `testtabs`. Tā satur ir kolonnas, kura nosaukumi ir vienādi arī ar izsaucāma mainīga nosaukumu.



```
1 <?php
2
3 class TesttabsController extends Zend_Controller_Action
4 {
5     public function indexAction()
6     {
7         $this->view->title = "Connected to db";
8         $testtabs = new Application_Model_DbTable_Testtabs();
9         $this->view->testtabs = $testtabs->fetchAll();
10    }
11 }
```

2.16. att. TesttabsController.php kontrolleris

No vienas puses ir labi, ka programmas izstrādāšanā un datu bāzes pieslēgšanā iespējami varianti, tas parāda ietvara lokanību. Taču sākuma tas liekas dīvaini un rada papildus jautājumus. Labi, ka pastāv iespēja ar komandām radīt jaunu projektu, modeli, skatu. Tas paātrina procesu.

Symphony

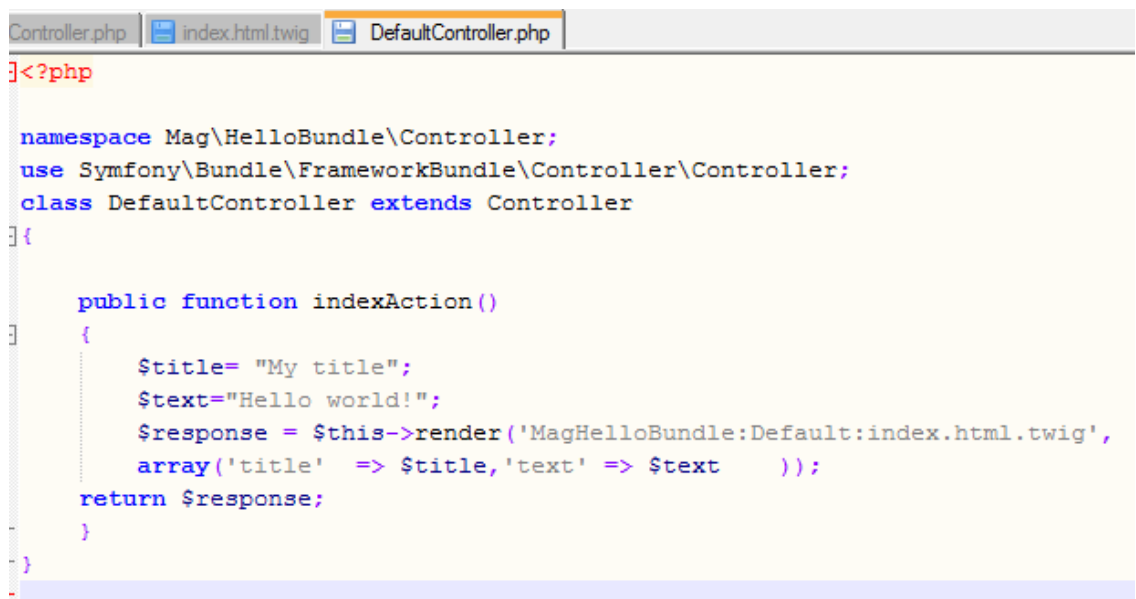
Programmu izstrādāšanai tika lietota Symfony izstrādātāju dokumentācija [18]. Sākumā tiek mēģināts izveidot „Hello world!” programmu.

Pirms sākt darbu ar kontrolleriem un skatiem, ir jāizveido pakete (bundle). Paketē Symfony2 glabā visu lietotnes kodu. Lai to izveidotu jāievada konmandrindā teksts: „`php app/console generate:bundle --namespace=Mag/HelloBundle --format=yml`”. Tālāk sistēma piedāvās apstiprināt nosaukumu, direktorijas un izveidos paketi. Šajā komandā „Mag” ir jebkurš vārds(izstrādātāji piedāvā ievest uzņēmuma vārdu), „HelloBundle” ir paketes nosaukums, tas paskaidro, ka tiks veidots lietotne „Hello world!”.

Izveidota lietotne tiek atrasts pēc url http://localhost/app_dev.php/hello

Atšķirībā no augstāk apskatītiem ietvariem, *Symfony2* ne vien seko attiecīgai failu struktūrai, pēc kuras veidojas url saiti un pēc kuram tiek piesaistīti kontrolleri un skati, bet arī pieraksta maršrutus atsevišķajos failos. Līdz ar to, ar vienkārši pakešu izveidošanu nepietiek, dažreiz maršruti ir jākorrigē `routing.yml` failos.

Katrā paketē ir mape ar kontrolleriem. Zemāk ir attēlots kā ir jāveido kontrolleris (sk. 2.17. att.). Funkcija `index()` atgriež definētas kontrollerī vērtības skatā `index.html.twig`. Šāda veida maršrutizācija un ceļu pierakstīšana apgrūtina procesu.

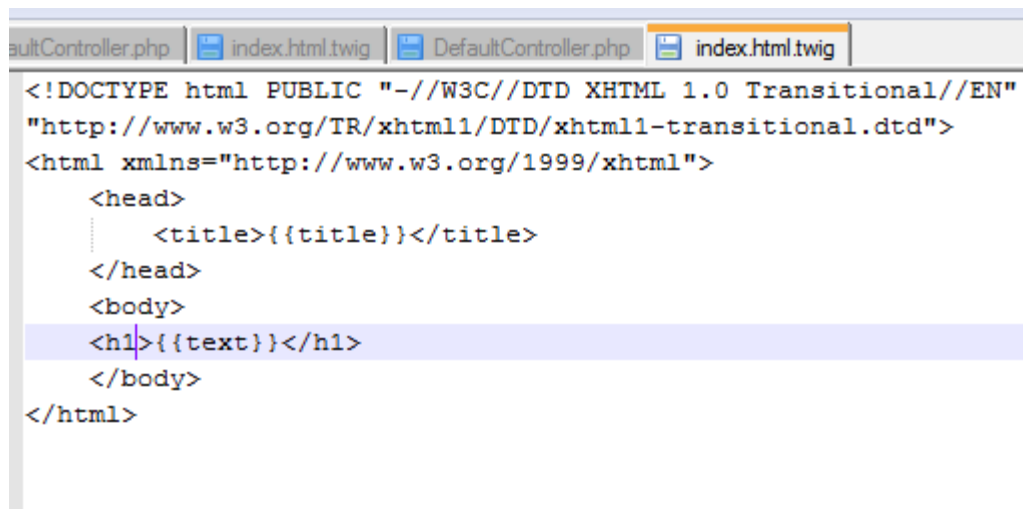


```
<<?php
namespace Mag\HelloBundle\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
class DefaultController extends Controller
{
    public function indexAction()
    {
        $title= "My title";
        $text="Hello world!";
        $response = $this->render('MagHelloBundle:Default:index.html.twig',
            array('title' => $title, 'text' => $text    ));
        return $response;
    }
}
```

2.17. att. `DefaultController.php` kontrolleris

Skatiem *Symfony2* izmanto Twig šablonizatoru (sk. 2.18. att.). Twig ir ātrs un salīdzinājuma ar vienkāršo PHP kodu, izmanto mazāk atmiņas. Tas ir orientēts uz šabloniem un saīsina koda daudzumu. Pie tam, pēc izstrādātāju domām, tas ir ērts, ātri apgūstams un drošs.

„{{title}}” nozīmē, ka šaja vieta tiks izmantots kontrollerī definēts mainīgais \$title, tas pats arī ar „{{text}}”. Ja šaja vietā izmantotu PHP, ne Twig, pieraksts būtu garāks, piemēram, „<?php echo \$title;?>”.



```
DefaultController.php | index.html.twig | DefaultController.php | index.html.twig
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>{{title}}</title>
  </head>
  <body>
    <h1>{{text}}</h1>
  </body>
</html>
```

2.18. att. **index.html.twig** skats

Darbs ar datu bāzi

Lai sāktu darbu ar datu bāzi, sakuma ir jāpiestrādā ar programmu Doctrine. Tā ir tehnoloģija, kas sasaista datu bāzi ar programmu (Object Role Model), atvieglojot darba procsu. Pielieto konsolī, lai izpildītu komandas. [21].

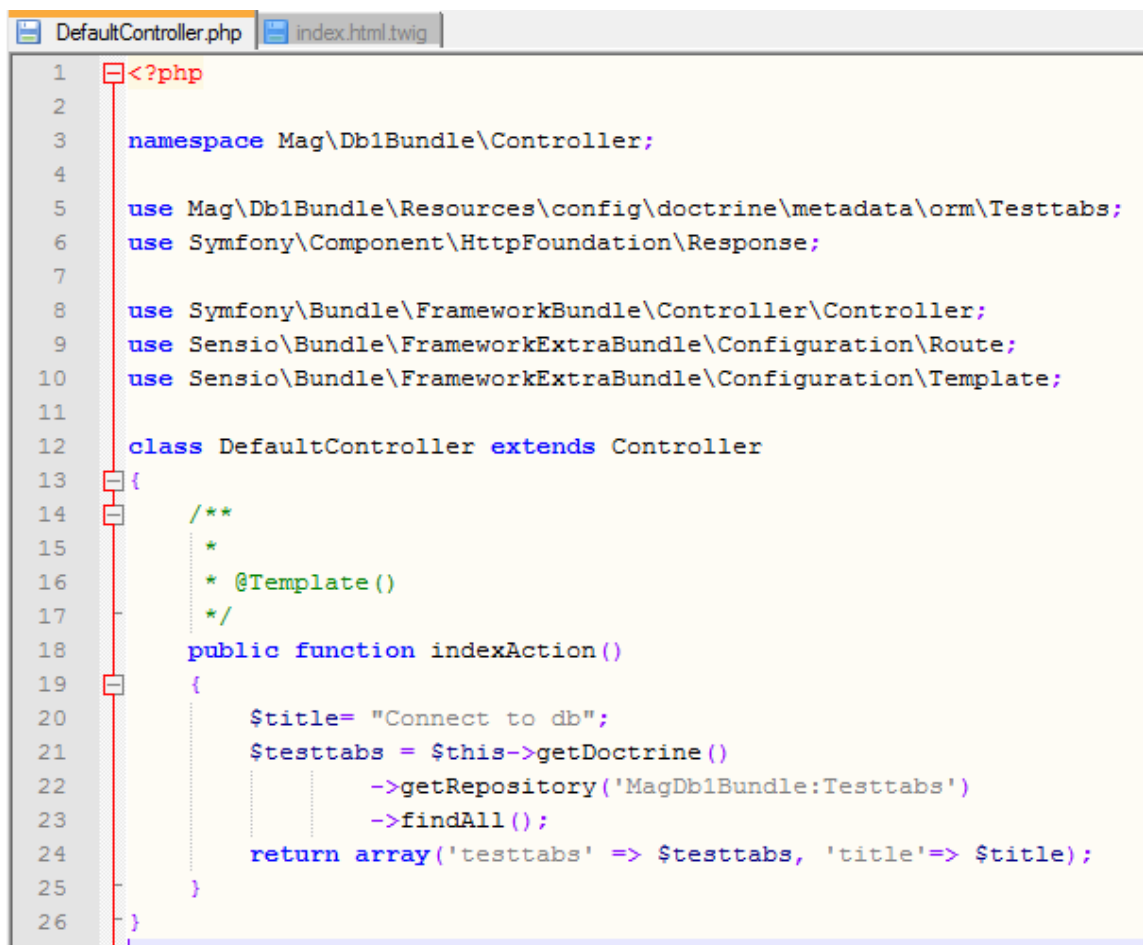
Meklējot paskaidrojumus darbam ar datu bāzi, Symfony2 liek uzsvaru darbam bez gatavas datu bāzes. Tas nozīmē, ka izstrādātāji daudz laika velta datu bāzes tabulu un lauku izveidošanai un pievienošanai, ka arī tālāko darbu skaidro pēc šāda uzskata. Taču veidojot programmu ar jau gatavu datu bāzi, radād grūtības, jo ir atšķirības.

Grāmatā „The Cookbook” [23] ir atrodams raksts par darbu ar esošo datu bāzi. Šim ietvaram nepietiek ar datu bāzes pieslēgšanu, ir nepieciešams veikt papildus darbības, lai paskaidrotu struktūru un lai būtu iespējams strādāt ar tabulām.

Sākumā ir jāizveido entītiju klases no esošas datu bāzes, pasakot „Doctrine” konsolei apskatīt esošu informāciju un neģenerējot metadatus failā. Metadati ir faili, kas paskaidro klasei ģenerēt atbilstošus laukus. Lai to sasniegtu, komandrindā ir jāievada komanda :” php app/console doctrine:mapping:convert xml ./ src/ Mag/ DbBundle/ Resources/ config/ doctrine/ metadata/orm --from-database --force” . Sī komanda noģenerē XML failus src/Mag/DbBundle/Resources/config/doctrine/metadata/orm mapē.

Pēc tā, komandām:” php app/console doctrine:mapping:import MagDbBundle annotation” un

„ php app/console doctrine:generate:entities MagDbBundle” var ieimportēt shēmu un veidot klases.



```
1 <?php
2
3 namespace Mag\Db1Bundle\Controller;
4
5 use Mag\Db1Bundle\Resources\config\doctrine\metadata\orm\Testtabs;
6 use Symfony\Component\HttpFoundation\Response;
7
8 use Symfony\Bundle\FrameworkBundle\Controller\Controller;
9 use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
10 use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
11
12 class DefaultController extends Controller
13 {
14     /**
15      *
16      * @Template()
17      */
18     public function indexAction()
19     {
20         $title= "Connect to db";
21         $testtabs = $this->getDoctrine()
22             ->getRepository('MagDb1Bundle:Testtabs')
23             ->findAll();
24         return array('testtabs' => $testtabs, 'title'=> $title);
25     }
26 }
```

2.19. att. DefaultController.php kontrollēris

```
DefaultController.php | index.html.twig
1 <!DOCTYPE HTML>
2 <html lang="ru-ru">
3 <head>
4   <meta charset="UTF-8">
5   <title>{{title}}</title>
6 </head>
7 <body>
8 <table>
9
10     {% for testtab in testtabs %}
11     <tr>
12     <td> {{ testtab.id }}</td>
13     <td> {{ testtab.coll }}</td>
14     </tr>
15     {% endfor %}
16
17 </table>
18
19 </body>
20 </html>
```

2.20. att. index.html.twig skats

Kā ir redzams kontrollerī (sk.2.19. att.) un skatā (sk.2.20. att.), koda daudzums ir lielāks, nekā iepriekšējos ietvaros. Pie tam, datu attēlošanai skatā, jāiepažīstas arī ar *Twig* šablonizatoru.

CakePHP

„Hello world” programmas izstrādāšana ar *CakePHP* gāja daudz ilgāk nekā ar CodeIgniter. Dokumentācija [13] piedāvā daudz piemērus, kā izveidot lappusi ar datiem no datu bāzes. Līdz ar to veidojot kontrolleru un skatu, programma izvada kļūdu, ka modelis nav atrasts (sk. 2.21. att.).

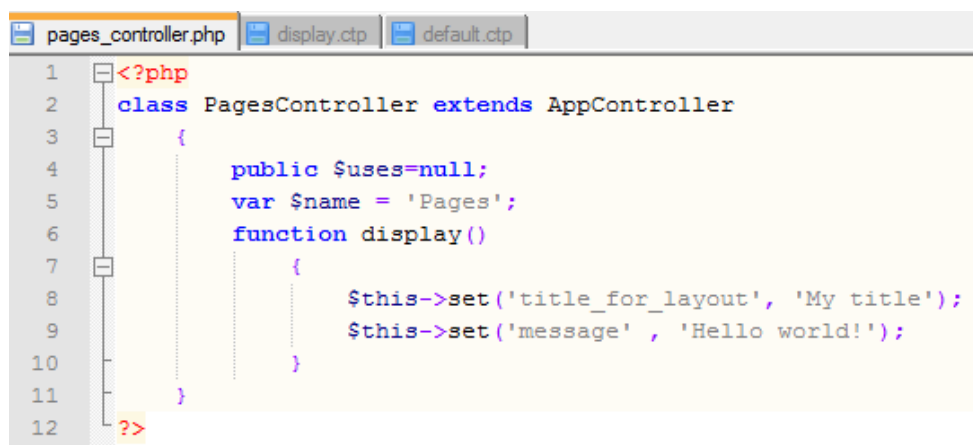
Missing Database Table

Error: Database table *pages* for model *Page* was not found.

2.21. att. CakePHP kļūdas ziņojums

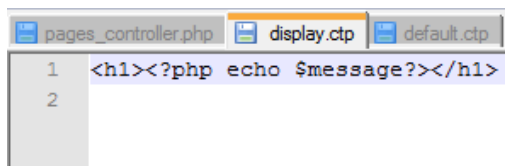
Dokumentācijas autori statisko lapušu izveidošanai iesaka lietot views/pages direktoriju, bet izstrādājot programmu šādā veidā, vienalga parādās kļūdas paziņojums. Problēmas risinājums netika atrasts ietvara dokumentācijā. Lai programma strādātu bez modeļa, (pieslēgšanaāš datu bāzei), kontrollerim jāpievieno komanda „public \$uses=null;”, kas pasaka , ka tabulas nav jāmeklē.

Kontrolleris pages_controller (sk. 2.22. att) tiek glabāts mapē app\controllers, bet skats display.ctp (sk. 2.23. att.) – app\views\pages. Rezultātu redzam pēc url <http://localhost/index.php/pages> , kur „pages” ir kontrollera vārds. *CakePHP* atver vajadzīgas lapas pēc likuma adrese/controlleras vārds/ metode.



```
1 <?php
2 class PagesController extends ApplicationController
3 {
4     public $uses=null;
5     var $name = 'Pages';
6     function display()
7     {
8         $this->set('title_for_layout', 'My title');
9         $this->set('message' , 'Hello world!');
10    }
11 }
12 ?>
```

2.22. att. pages_controller.php kontrolleris



```
1 <h1><?php echo $message?></h1>
2
```

2.23. att. display.ctp skats

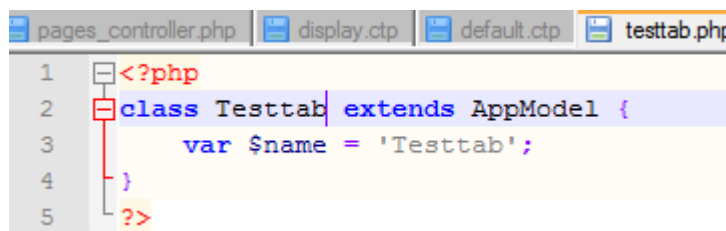
Pēc noklusējuma *CakePHP* pievieno katrai izveidotai lappusei kājeni un galveni (header footer). Lai to noņemtu ir jāizveido jauns šablona ar lapas struktūru, kur `$content_for_layout` būs tas, ko satur kontrolleris, bet `$title_for_layout` ir lapas virsraksts (definējams kontrollerī). Pie tam, veicot ātruma testus, standarta header un footer lādēšana aizņem arī papildus laiku. Eksistē pamata šablons, kas nosaka kur tiks parādīts skata saturs, ka arī skata šablona, kas konkretizē izvadāma teksta izskatu. Pamata šablons ir ļoti noderīgs, jo tas ekonomē vietu un ļauj neatkārtot vienu un to pašu katrā skatā.

Rezultatā vienkāršas programmas izstrādāšana aizņēma daudz vairāk laika, nekā izmantojot CodeIgniter un citus ietvarus.

Darbs ar datu bāzi.

CakePHP pieprasa arī savu tabulu nosaukumu likumu. Tas nozīmē, ka tabulas nosaukumam jābūt daudzskaitlī – testtabs.

Sākumā jāizveido modelis, kas atbilst datu bāzei. Autore izmanto tabulu testtabs. Nosaukumiem ir liela nozīme *CakePHP*. Nosaucot mūsu modeli testtabs, *CakePHP* automātiski sapratīs, ka šis modelis atbilst kontrollerim testtabs_controller un tas ir piesaistīts tabulai testtabs. Modelis glabājas direktoriņā `app\models` (sk. 2.24. att.).



```
1 <?php
2 class Testtab extends AppModel {
3     var $name = 'Testtab';
4 }
5 ?>
```

2.24. att. testtab.php modelis

Nākamais solis ir kontrollera izveidošana (sk. 2.24. att.). To saglabājam `/app/controllers` direktoriņā. Kontrollerī norādam lapas virsrakstu un norādam, ka jāsaņem visi dati no tabulas un jāglabā to mainīgajā testtabs.

```
pages_controller.php | display.ctp | default.ctp | testtab.php | testtabs_controller.php | in
1 <?php
2 class TesttabsController extends ApplicationController {
3     var $name = 'Testtab';
4     function index() {
5         $this->set('title_for_layout', 'Connect to db');
6         $this->set('testtabs', $this->Testtab->find('all'));
7     }
8 }
9 ?>
```

2.25. att. pages_controller.php kontrollieris

Pēdējais ir skata izveidošana (sk. 2.26. att.). Tā ka šablons ar sākuma *HTML* tegiem jau bija izveidots darba ar „Hello world” lietotni, pievienojam nepieciešamus *HTML* tegus un noradam kura vieta jāieliek virsraksts un datu bāzes dati.

```
s_controller.php | display.ctp | default.ctp | testtab.php | testtabs_controller.php | index.ctp
<table>
  <?php foreach ($testtabs as $testtab): ?>
  <tr>
  <td><?php echo $testtab['Testtab']['Id']; ?></td>
  <td><?php echo $testtab['Testtab']['col1']; ?></td>
  </tr>
  <?php endforeach; ?>
</table>
```

2.26. att. index.ctp skats

Programma tiek atvērta pēc url <http://localhost/index.php/testtabs>. Viss strāda ar pirmo reizi. Tas nozīmē, ka ir diezgan grūti izveidot pirmo vienkāršo programmu, tāpēc, ka dokumentācijā tas nav aprakstīts. Taču nākamie soļi ir izpildāmi ātri. Darbu ar datu bāzi dokumentācija apraksta diezgan labi.

Yii Framework

Programmu izstrādāšanai tika izmantota lietotāja pamācība [16]. „Hello world!” programmas izstrādāšanai bija jāizveido kontrolleris, skats un šablons to attēlošanai, līdzīgi, ka programmas veidošanai ar *CakePHP*. Lai atvērtu lapu ar programmas saturu, jāuzraksta sekojoša adrese: <http://hostname/index.php?r=hello/index> (vai <http://hostname/index.php?r=hello>), kur „hello” ir kontrollera nosaukums HelloController.php (sk. 2.27. att.), index ir darbība. Sākumā tika izveidots kontrolleris, kura definējam divus mainīgus un nosakam, kuram skatam tie ir domāti.

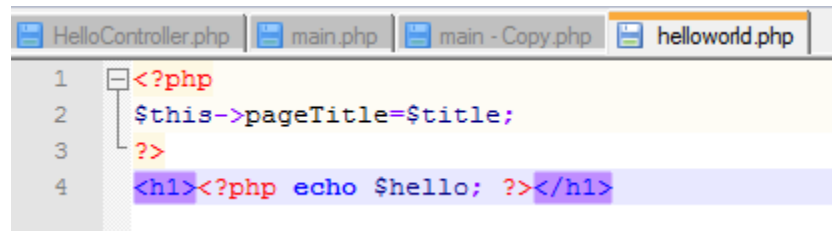


```
1 <?php
2 class HelloController extends CController
3 {
4     public function actionIndex()
5     {
6         $message['title'] = 'My title';
7         $message['hello'] = "Hello World!";
8         $this->render('helloworld', $message);
9     }
10 }
11 ?>
```

2.27. att. HelloController.php kontrolleris

Nekādus ierakstus, ka programmai nevajadzētu vērsties pie datu bāzes nav jāraksta (ka *CakePHP*). Pēc autores domām, tas ir labi. Nākamais solis ir šablona izveidošana (sk. 2.28. att.). Šablons tiek izveidots tads pats, kā *CakePHP* piemērā. Tas ir diezgan ērti, jo gadījumā, kad tīmekļa vietnei ir vienāds izskats vairākām lapām, tas ir ērts risinājums. \$content daļā, tiek parādīta visa skata esoša informācija. pageTitle arī varēs mainīt, to vērtību ir jānorāda kontrollerī, bet skatā jāpasaka, ka šablona virsraksts būs jāmaina uz izvēlēto.

Ar komandu \$this->pageTitle=\$title programma saprot, ka ir jāmaina šablonā definētais virsraksts. \$hello izvada uz ekrāna kontrollerī ievadītu tekstu.



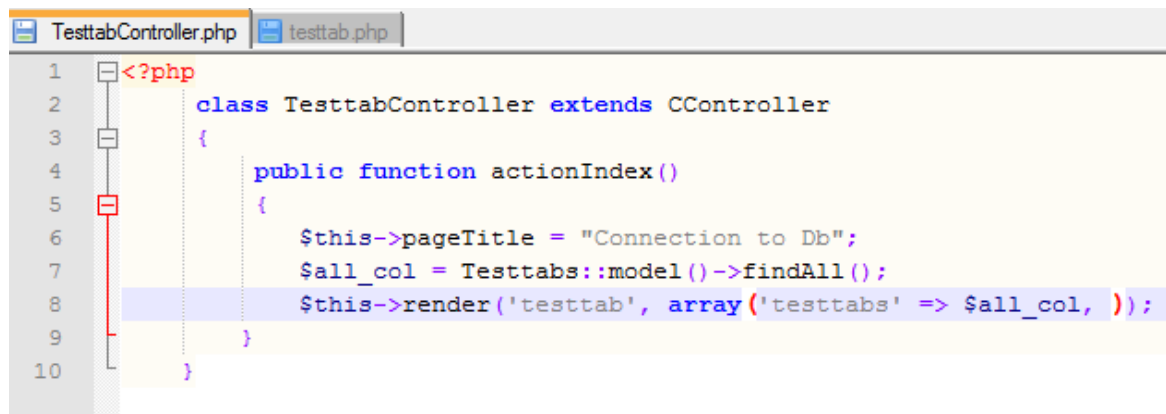
```
1 <?php
2 $this->pageTitle=$title;
3 ?>
4 <h1><?php echo $hello; ?></h1>
```

2.28. att. main.php skats

Programmas izstrādāšana ir diezgan viegla, saprotama un loģiska. Ietvara izmantojamās metodes ir aprakstītas dokumentācijā, aprakstos par kontrolleriem un skatiem ir piedāvata nepieciešama informācija. Interneta atrodams daudz papildinformācijas. Dokumentācija izstrādāta labi un saprotami. Programmas izstrādāšana ar *Yii* bija daudz vieglāka, nekā ar *CakePHP*, bet apmēram vienāda ar *CodeIgniter*.

Darbs ar datu bāzi.

Otras programmas izstrādāšana bija ļoti viegla un intuitīvi skaidra. Kontrollerī (sk. 2.29. att.) autore definēja lapas nosaukumu un mainīgo, kas saturēja datus no norādīta modeļa (tabulas). Abus mainīgos virzīja uz norādīto skatu.



```
1 <?php
2 class TesttabController extends CController
3 {
4     public function actionIndex()
5     {
6         $this->pageTitle = "Connection to Db";
7         $all_col = Testtabs::model()->findAll();
8         $this->render('testtab', array('testtabs' => $all_col, ));
9     }
10 }
```

2.29. att. TesttabsController.php kontrolleris

Modelis tika izveidots patstāvīgi, kaut gan programma i pastāv iespēja ģenerēt to kopā ar pamatiespējām automātiski. Tas ļauj veikts tādas darbības kā ierakstu atjaunošana, apskatīšana, dzēšana, pievienošana jau gatavajā interfeisā. Ļoti ērti un ekonomē laiku. Ta ka šajā programmā nav nepieciešams veikt augstāk minētas darbības, izņemot datu apskatīšanu, modelis izskatās ka 2.30. attēlā.

```
1 <?php
2
3     class Testtabs extends CActiveRecord
4     {
5         public static function model($className=__CLASS__)
6         {
7             return parent::model($className);
8         }
9         public function tableName()
10        {
11            return 'Testtabs';
12        }
13    }
14 >>
```

2.30. att. Testtabs.php skats

Šablons netiek mainīts. Skats arī ir izveidojams elementāri (sk. 2.31. att.).

```
1 <table>
2 <?php
3     foreach($testtabs as $row)
4     {
5         echo "<tr>";
6         echo "<td>".$row['Id']."</td>";
7         echo "<td>".$row['coll']."</td>";
8         echo "</tr>";
9     }
10 >>
11 </table>
```

2.31. att. pages_controller.php kontrolleris

Pozitīvs vērtējums dokumentācijai un izstrādāšanas procesam.

Secinājumi

CodeIgniter programma strādāja ar pirmo mēģinājumu. Dokumentācija ļoti skaidri un īsi paskaidro visu galveno, parāda ar piemēriem. Darbs ar datu bāzi arī bija skaidrs. Nekādas darbības ar modeli neprasīja. Failu glabāšanas struktūra skaidra, darba uzsākšana vienkārša.

Pēc autores domām, *Zend Framework* ir vairāk draudzīgs ar pieredzējušākiem lietotājiem, jo dokumentācija bieži vien nav labākais ietvara darbības „skaidrotājs”. Tomēr pirmās programmas izstrādāšana ir viegla. Lai strādātu ar datu bāzi, atšķirība no *CodeInteger*, ir jāveido arī modeli, kur jāieraksta darbības. Faili tiek veidoti pēc komandas, taču jāizpilda tie patstāvīgi. Kopuma darbs ar šo ietvaru atstāj labas atsauksmes.

Symfony programmas nebija tik ātri paveiktas. Jādara daudz darbības, kontrollera saturs ir sarežģītāks un garāks. Strādājot ar *Symfony*, ir labāk izpētīt arī taja izmantotu šablonizatoru *Twig* un, darbam ar datu bāzi nepieciešamu- *Doctrine*.

Pirmās programmas izstrādāšana ar *CakePHP* gāja daudz ilgāk nekā ar *CodeIgniter*. Tas notika, jo darbam netika izmantota datu bāze, bet kontrolleris automātiski meklēja modeli. Toties darbs ar datu bāzi ir viegls un saprotami dokumentēts.

Autoresm īpaši patika darbs ar *Yii framework*. *Yii* izstrādātāji parūpējas par iespēju automātiski ģenerēt modeli ar pamata darbībām. Tādas darbības, kā dzēst, pievienot, apskatīt un labot ir pamatdarbības, kuras vajadzīgas jebkuram lapas administratoram. Līdz ar to darbs ar datu bāzi un modeļiem ir ļoti patīkams un viegls. Failu struktūra saprotama un viegla, programmām nepieciešamu failu izstrādāšana intuitīvi skaidra un nesagādāja nekādas grūtības, šablonu izmantošana laba. Programmas izstrādāšana ar *Yii* bija daudz vieglāka, nekā ar *CakePHP*, bet apmēram vienāda ar *CodeIgniter*.

Programmu izstrādāšana bija labāka *CodeIgniter* un *Yii* ietvariem. Taču *Yii* ir vērtēts augstāk, tā labāka darba ar datu bāzi dēļ.

2.2.4.Ātrums

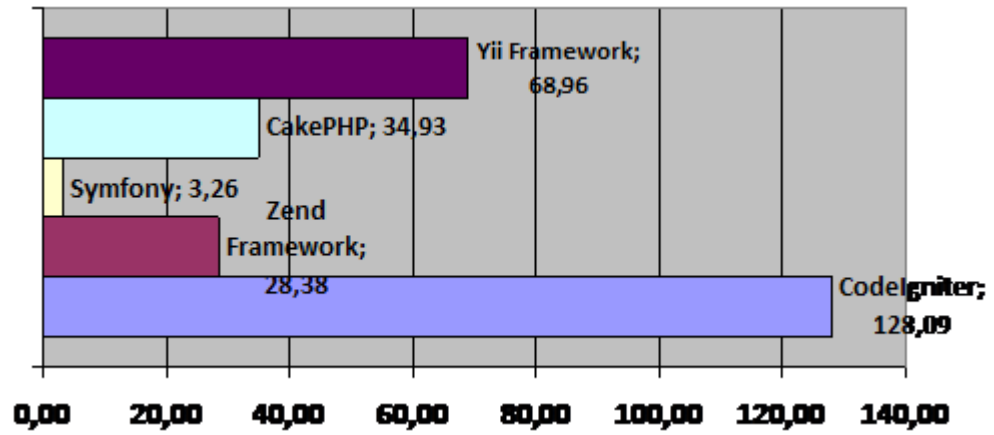
Testēšanai tiek izmantots *ApacheBench*, kas ir aprakstīts 3.2. nodaļas sākumā. Autore piedāvā apskatīt paša iegūtus rezultātus, kā arī iepazīties ar *Yii* komandas izstrādātāju veiktajiem testiem.

Autores pieeja ir mērīt abas programmas ar *MVC* struktūras izmantošanu, kad kontrolleris nodos vērtības skatam, bet tad izvada rezultātus. Attēlā 2.32. ir apskatīts variants ar „Hello world!” komandu, bet attēlā 2.33. darbs ar datu bāzes tabulu. Autore testē bez PHP paātrinātājiem (*APC*), tie nav obligāti pielietojami, tikai *Symfony* izstrādātāji iesaka lietot kādu paātrinātāju, taču, lai nosacījumi būtu visiem vienādi, mērījumi ir veikti bez *APC*. *Yii* piedāvā abus variantus (sk. 2.34. att.).

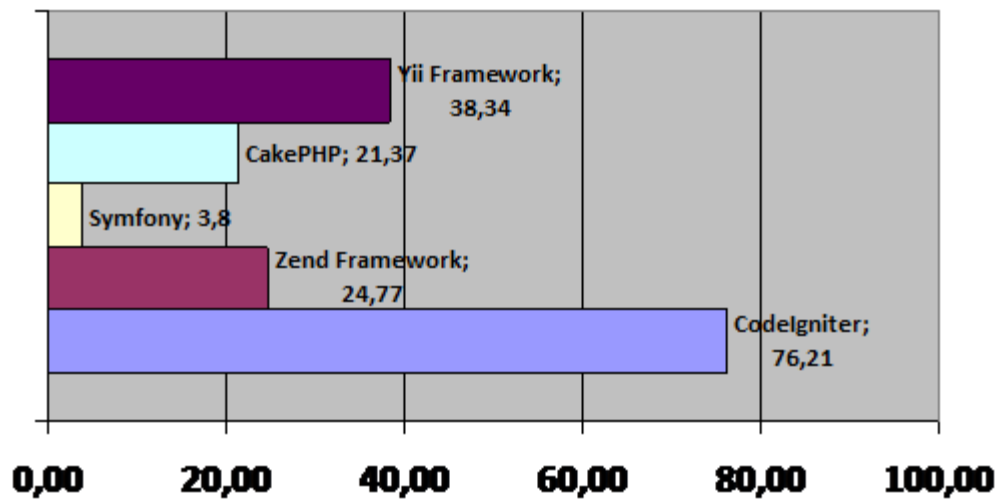
Yii komanda apskata ātrumu, izstrādājot tikai die(). Tas nozīmē, ka *Yii* komandas rezultāti var būt precīzāki, taču autores veiktie pētījumi ir tuvāki reālai dzīvei.

Rezultāti ir attēloti pieprasījumos sekundē (*Requests Per Second, RPS*), tāpēc, lielākas vērtības ir labākas.

Rezultāti tiek attēloti tabulā xx. Visvairāk punktu saņem ātrākie ietvari, pārējie, atkarībā no iegūtās vietas, saņem savus vērtējumus. Tā ka *CakePHP* un *Zend Framework* rezultāti ir mainījušies vietām testu gaitā, bet otrajā testā tie ir vienādi, tiem tiek piešķirta vienāda 3.vieta. *CodeIgniter* un *Yii* ir līderi gan autores, gan *Yii* komandas izstrādātāju testos, bet tie ir mainījušies vietām. Tas nozīmē, ka ātrākie iegūs 5 punktus, bet lēnākais 2.

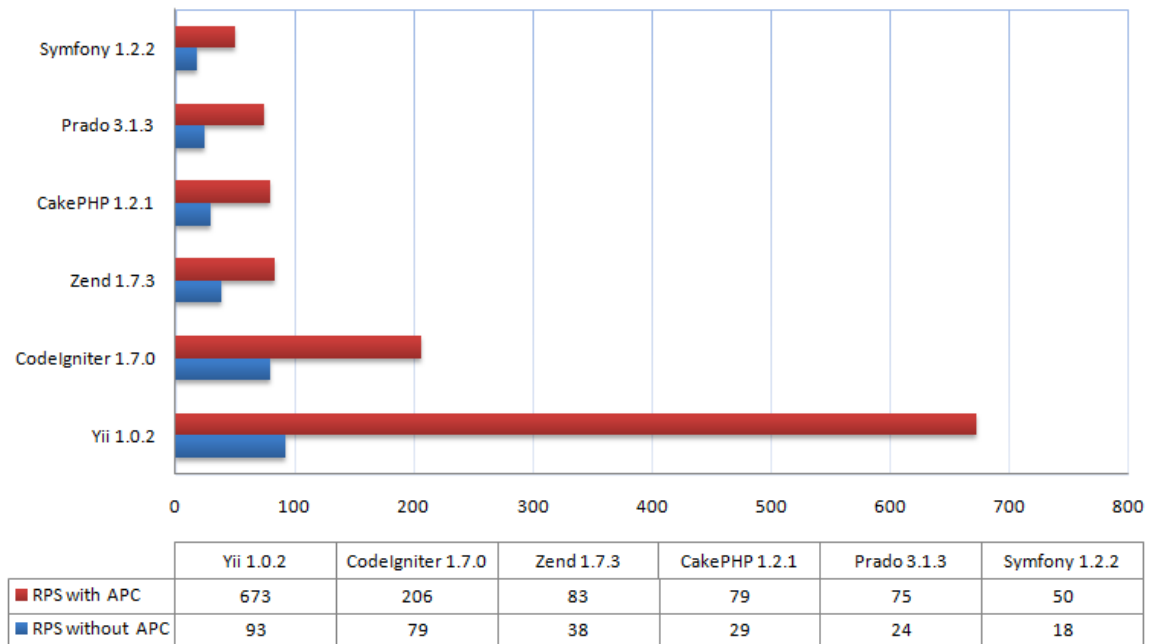


2.32. att. Autores ātruma testa „Hello world” programmai rezultāti



2.33. att. Autores ātruma testa darbam ar datu bāzi rezultāti

PHP Framework Performance Comparison



2.34. att. Yii Framework komandas ātruma testa rezultāti

Secinājumi

Pēc autores veiktajiem testiem, ātrākais ir *CodeIgniter*, otrajā vietā ir *Yii framework*. Paskatoties uz Yii komanda izstrādātāju testa, līderi ir tie paši, bet mainītie ar vietām. Abiem tiek piešķirti uzvarētēju 5 punkti.

2.2.5.Drošība

Šajā nodaļā tiks apskatīti ietvaru izstrādātāju komponentes, vai rīki, kas domāti aktuālajiem vietnes drošības pasākumiem. Pamatprasības ir šādas:

- 1.Autentifikācija/autorizācija- rīks, kas nodrošina šo procesu darbību. Interneta veikalā tiks veidota lietotāju sistēma, kura būs nepieciešama reģistrācija un autorizācija.

2. XSS filtrācija (no angļu valodas Cross Site Scripting), filtrs meklē tehnikas, kas ir izmantojamas bīstama kodu meklēšanai (sessijas datu zagšanai utt.);
3. Datu pārbaude (validācija)- ievadāmo datu pārbaude, lai tie atbilstu noteiktam tipam. Interneta veikalā ir paredzētas formas un filtri, kas paredz izmantot teksta laukus.
4. SQL aizsardzība- ievadīto datu filtrācija, kas aizliedz, attīra simbolu virkni no aizliegtiem simboliem, kuru izmantošana var novest pie datu bāzes bojāšanas (sql-injection).
5. *Captcha*- ir tehnoloģija, kas pārbauda, vai sistēmas lietotājs ir cilvēks vai mašīna. Piemēram, uz ekrāna tiek izvadīts attēls ar simboliem, kurus ir jāieraksta attiecīgajā laikā. Tiek uzskatīts, ka tas ir neizpildāms datoram, bet viegli veicams cilvēkam. Lai apgrūtinātu robotu ietekmi.

Ja ietvara izstrādātāji piedāvāja arī kādus citus, interesantus pasākumus, tie arī tika apskatīti.

CodeIgniter

CodeIgniter piedāvā šādas funkcijas augstāk minēto darbību izpildei:

1. *Zend_Acl*, *Ion Auth* – reģistrācijai un autorizācijai;
2. *Input* klase, kas nodrošina XSS filtrāciju;
3. *Form Validation* klase – ievadāmo datu pārbaudei.
4. *Database* klase ļauj izmantot datu pārbaudi, tiek piedāvatas komandas datu pārbaudei no neatļautajiem simboliem;
5. *CAPTCHA Helper*: palīgs darbam ar captcha, tas piedāvā iespēju izmantot nepieciešamas funkcijas.

Pirmais ieteikums, ko piedāvā Codeigniter izstrādātāji ir pārsaukt „system” mapi un norādīt tās jaunu vārdu failā `index.php` mainīgajā `$system_folder`.

Codeigniter izstrādātāji apraksta ietvara drošības pasākumus dokumentācijā, kur labākie risinājumi sadalīti pa punktiem:

URI drošība – *CodeIgniter* ierobežo simbolus, kuri var būt izmantoti *URI* rindiņā, lai minimizētu bīstamu datu sūtīšanu lietotnei. *URI* var saturēt tikai sekojošu: alfabēta- ciparu simbolus, „~”, „.”, „:”, „_”, „-”.

Register_globals – inicializējot sistēmu, tiks iznīcināti visi globāli mainīgie, izņemot masīvus \$_GET, \$_POST un \$_COOKIE.

error_reporting – tiek piedāvāts izslēgt PHP kļūdu izvadu, lai kļūdas nevarētu atklāt potenciāli konfidenciālu informāciju. Uzsādot konstantei ENVIRONMENT, kura atrodas index.php failā ar vērtību 'production', secinās šo kļūdu izvadu uz ekrāna. Izstrādāšanas brīdī vēlams šo vērtību atstāt uz 'development'.

magic_quotes_runtime – šī direktīva tiek atslēgta sistēmas inicializācijas brīdī, tāpēc nav obligāti dzēst „/” saņemot informāciju no datu bāzes.

Tika atrasta arī vērtīga Encryption klase, kas šifrē datus. [6].

Zend Framework

- 1.Zend_Auth: autentifikācijas komponents;
- 2.Cross Site Scripting (XSS) Protection;
- 3.Zend_Validate – ievadāmo datu pārbaudei;
4. SQL Security – SQL aizsardzībai;
- 5.Zend_Captcha;

Zend Framework piedāvā aizsardzību pret CSRF (angļu Cross Site Request Forgery), lietotāju aizsardzība, kad ļaundari izmanto HTTP protokolu vājās puses, piemēram, sūta pieprasījumus uz citu tīmekļa vietni, kura slepeni veic kādas darbības (operācijas ar naudu).

Symphony

- 1.Firewalls (autentifikācija-pārbaude) un Access Controls (autorizācija-piekļuves tiesības): ietvars piedāvā veikt lietotāja datu pārbaudi un dot piekļuves tiesības. Veikamās darbības aprakstītas dokumentācijas grāmatā sadaļā security [18].

2. Pec Symfony izstrādātāju bloga apskatīšanas, tika atrasts raksts, ka Symfony 1.2 versijā toka novērstas drošības trūkumi XSS ziņā. Ta ka sākot ar šo versiju arī šajā ziņā problēmu nav. Speciāli moduļi, jeb rīki nav atrasti [23].
- 3.validator Service klase – ievadāmo datu pārbaudei;
- 4.Doctrine atbildīga par aizsardzību SQL injekcijām;
5. sfFormExtraPlugin- satur sevī vairākas rprogrammas un rīkus proekš formām, tai skaita arī *CAPTCHA*. [18]
- 6.Symfony piedāvā vairākas grāmatas ar dokumentāciju dažādiem nolūkiem, pie tam eksistē arī izstrādātāju blogs[18, 22, 23], kur tiek anonsēti jaunumi un izmaiņas, kā arī atbildes uz dažiem jautājumiem. Tāpēc, lai atrastu vajadzīgu rīku, ir jāmeklē visur.

CakePHP

CakePHP šādas bibliotēkas *Authentication* un *Access Control Lists* risina autentifikācijas un autorizācijas problēmu. Esoša Security komponente, satur iebūvētus rīkus , lai validētu datus no formas, xss aizsardzību, drošību pret *SQL* injekcijām, arī aizsardzību pret *CSRF* (angļu Cross Site Request Forgery), lietotāju aizsardzība, kad ļaundari izmanto HTTP protokolu vājās puses, piemēram, sūta pieprasījumus uz citu tīmekļa vietni, kura slepeni veic kādas darbības (operācijas ar naudu) [24].

Captcha komponents nav iebūvēts, neeksistē *CakePHP* bibliotēkā, bet to var ieintegrēt, pamācība <http://bakery.cakephp.org/articles/stephanoff/2007/01/14/integrate-cakephp-with-kcaptcha>.

Yii Framework

Tā ka *Yii framework* jau starta piemēra parāda lietotājam kā izskatās formas, lapas, navigācija, tad iekšā jau ir iebūvēta autentifikācija. Bet pie tam *Yii* izstrādātāji piedāvā ļoti labu un ērtu pielikumu *yii-user* [25]. Tas nodrošina daudzlietotāju piekļuvi vietnei, tiek izmantota datu bāze, ir iespējama autorizācija un daudzlīmeņu piekļuves tiesību sistēma. Pieslēgšana notiek ļoti saprotami.

HTMLPurifier komponente rūpējas par *XSS* uzbrukumiem [26];

Validācija arī ir izmantojama un definēta kontrolerī. Pie tam definēt var ne vien saņemamo datu tipu un maksimālo/minimālo garumu, bet arī laukus, kas ir obligāti, vai lauki, kuru ievadītais teksts jāsalīdzina ar citu tekstu [18].

Lai parūpētos par SQL drošību pret injekcijām, Yii izstrādātāji piedāvā izmantot izteiksmju sagatavošanu ar parametru marķieriem (placeholders), kuri darbību rezultāta būs mainīti uz reāliem parametriem.

Piemēram, *SQL* izteiksme ar :username un :email marķieriem:

```
$sql="INSERT INTO tbl_user(username, email) VALUES(:username,:email)";
```

```
$command=$connection->createCommand($sql);
```

Šāda *SQL* aizsardzība ir drošāka par ievadīta teksta pārbaudi ar standartra funkcijām, kas aizliedz speciālus simbolus, jo šādi nav jāpielieto pārbaudes katram ievadparametram un ir garantija, ka nebūs atrasts kāds neaizliegts simbols, kas būs bīstams.

Ir *Captcha* elements.

Yii piedāvā arī aizsardzību pret CSRF [26].

Komponenšu pieslēgšana notiek viegli, piemēram:

```
'components'=>array(  
  
    'request'=>array(  
  
        'enableCsrfValidation'=>true,    ),    );
```

Secinājumi

Autorizācija ir īstenota visos ietvaros, vairākums izstrādātāju parūpējās arī par reģistrāciju. *Yii Framework* ietvara izstrādātāji piedāvā iebūvētu autorizāciju, kur pašos failos ieraksta piekļuves informāciju un lietotājus, kā arī paplašinājumu *Yii-user*, kas uztur pilnu reģistrācijas un autorizācijas procesu, kā arī lietotāju administrēšanu.

Par XSS filtrāciju ir padomājuši visi izstrādātāji, visi piedāvā klases, jeb komponentus, kas apsola drošību šajā ziņā. Datu pārbaude – validācija arī ir veicama visur.

SQL aizsardzība pret injekcijām, ir realizēta visur, bet Yii Framework izmanto piedāvā izmantot izteiksmju sagatavošanu ar parametru marķieriem (placeholders), kas garantē drošību.

Captcha – tehnoloģija, kas pārbauda, vai sistēmas lietotājs ir cilvēks vai mašīna ir pieejama visos ietvaros, to vai nu ir jāpieslēdz, vai jāpievieno.

2.2.6. Dokumentācija

CodeIgniter

Izstrādājot vienkāršu lietotni “Hello world”, kā arī izstrādājot lapas versiju ar datu bāzes izmantošanu, autore secināja, ka dokumentācija ir ļoti labi izstrādāta. Visas darbības, kas jāveic no paša sākuma, līdz programmu izveidošanas ir skaidri un saprotami aprakstītas dokumentācijā. Ir pievesti piemēri, kas palīdz apgūt apmācāmo, dokumentācijas raksti nav ne gari, ne īsi, nekā lieka, ka arī nekas netrūkst. Pēc autores domām, dokumentācija ir izstrādāta lieliski, tas veicina CodeIgniter pievilcību no lietotāja puses, sākot ar pirmiem soļiem.

Zend Framework

Dokumentācija ir izstrādāta diezgan labi. Bet pirmie soļi ir saprotami ne īpaši viegli. Pēc autores domām, piedāvātais piemērs, *Zend Framework* instalācijai un darbam nepieciešamas darbības, bija ne īpaši saprotami. Tiek piedāvāts ielādēt testa piemēru, lai redzētu failu struktūru. Toties ir pieejams diezgan daudz citu resursu, kas paskaidro ietvara darbību līdz sākumiem. Parāda komandas, kas atvieglo un automatizē projekta veikšanu.

Nepieciešama informācija ir pieejama, bet ta ir piedāvāta ļoti plašā veidā, informācija ir vajadzīga, bet grūti atrodamā. Jaunam lietotājam ir ērtāk pameklēt apmācību citos portālos. Taču zinot pamatdarbības, viss kļūst skaidrāks. Gribētots redzēt vēl sīkāku aprakstu ar sākumam nepieciešamām darbībām.

Symphony

Ar pirmo acu skatienu, liekas, ka dokumentācija nav pilnīga. Bet uzsākot pirmos soļus, viss iet ļoti saprotami un pareizi. Nerodas kļūdas un nesaprašanās. Uzstādīšanas stadija dokumentācija

tiek vērtēta augsti. Nav liekas informācijas, tikai nepieciešamais. Īsi, konkrēti un skaidri. Ir vairaki avoti, no kurienes smelt informāciju. Gan grāmatas, kuras var gan nopirkt, gan ielādēt .pdf formāta, gan blogs, gan pamācība oficiālajā vietnē. Pie tam, internetā ir atrodami papildus skaidrojumi.

CakePHP

Dokumentācija izstrādāta pārāk sīki, bet ne īsti informatīvi. Iespējams, ka pieredzējušam *CakePHP* lietotājam tas tā neliksies, bet pirmie soļi ir veicāmi diezgam grūti. Nav ērti lietojamas. Eksistē daudz pamācību, dažādās valodās, dažādas versijas. Bet meklējot vajadzīgo, jātērē daudz laika, bieži vien ta arī neatradot ko vajag. Sākuma grūti saprast kā strāda url veidošana, pēc autores domām, šādas lietas vajadzētu uzrakstīt programmas veidošanas sākumā. Visos piemēros, url tiek parādīt pēc sekojušas shēmas: `http://localhost/controller`, īstenībā, lai atvērtu vajadzīgu lapu klāt jāpievieno arī `index.php` `http://localhost/index.php/testtabs`, ko var noņemt. Ta ka programma ir uzstādīta pēc izstrādātāju rekomendācijas ar standarta nostādījumiem, arī rezultātu būtu jāgaida pēc standarta.

Dokumentācija ir laba pieredzējušajam lietotājam, pirmo reizi ar to strādāt ir grūti. Toties *CakePHP* piedāvā maksas kursus jebkura līmeņa lietotājiem.

Ne īpaši labu iespaidu atstāja stabilas jaunākas versijas *CakePHP* uzstādīšana, jo rezultātā pieslēgšanās pie datu bāzes nebija veikta. Bet iepriekšējās versija ar tādiem pašiem datiem strādāja korekti.

Ērtāka dokumentācija ir avotā 19.

Yii Framework

Yii ietvars, kā arī iepriekšējos punktos ir augsti vērtēts. Izstrādātāji piedāvā labu apmācību, gan grāmatas iesācējiem, kā arī pieredzējušākiem lietotājiem. Ir peejams arī *Wiki* resurss.

Dokumentācija palīdz saprast ietvara darbības specifiku, ta ir loģiska, Ir pievesti piemēri, kas palīdz apgūt apmācāmo, dokumentācijas raksti nav ne gari, ne īsi, nekā lieka, ka arī nekas netrūkst. Pēc autores domām, dokumentācija ir izstrādāta lieliski.

Secinājumi

CodeIgniter dokumentācija atstāja labu iespaidu. Neradās ne jautājumu, ne grūtību. Viss ļoti skaidri un saprotami paskaidrots.

Zend Framework dokumentācija, no nepieredzējoša ietvara lietotāja viedokļa, nav izstrādāta ļoti skaidri. Aizvien rodas jautājumi, atbildes uz kuriem, tiek labāk izskaidrotas citos resursos. Bet pēc programmu izstrādāšanas, dokumentācija jau liekas saprotama.

Symfony dokumentācija izstrādāta rūpīgi. Par lielu plusu tiek uzskatīts, ka ir vairaki avoti, no kurienes smelt informāciju.

CakePHP dokumentācija ir izstrādāta diezgan labi un saprotami.

Yii dokumentācija ļoti labi izstrādāta. Viss saprotams. Daudz informācijas un avotu.

Par labākiem dokumentācijas ziņā autore nosaka *CodeInteger* un *Yii framework*.

2.2.7.Citi

CodeIgniter

Par stipru *CodeIgniter* pusi autore uzskatīja Palīgus, kas izveidoti, lai atvieglotu vietnes izstrādāšanu programētājam. Katrs palīga fails ir vienkārša funkciju kolekcija attiecīgajā sadaļā. Ir Form palīgi, kas atvieglo darbu ar formām, Text palīgs, kas veicina dažādas teksta formatēšanas pasākumus, Cookie palīgi izveido un nolasa cookies, bet File palīgi piedāvā darbu ar failiem utt. (*Array, Cookie, Date, Directory, Download, Email, File, Form, HTML, Language, Path, Smiley, String, Text, URL, XML*) . *CodeIgniter* nelādē palīgus pēc noklusējuma. Tāpēc, lai tos izmantotu sākumā jāielādē no system/helpers un application/helpers. Ielādēti palīgi ir atrodami kontrolleros un skatos.

Codeigniter piedāvā arī vairākas vērtīgas klase, piemēram:

- Pagination Class- klase darbam ar lappusēm;
- klase Email piedāvā šādas iespējas:

- dažādu protokolu izmantošana (Mail, Sendmail un SMTP);
- vairāki saņēmēji;
- HTML* un Plaintext;
- pielikumi;
- vārdu pārvešana un citi.
- klase Cart ļauj pievienot elementus sessijā, kuru izmanto lietotājs, strādājot ar tīmekļa vietni. Elementi var būt attēloti standartā groza veidā, ļaujot lietotājam mainīt daudzumu un dzēst elementus no groza;
- klase Language- ļauj veidot failus ar sev nepieciešamu valodu, pārslēgt to pēc nepieciešamības.

Zend Framework

Zend izstrādātāji piedāvā veselu sarakstu ar risinājumiem, zemāk daži no tiem ir diezgan interesanti un negaidīti, bet var būt noderīgi:

- Zend_Translate- risinājums, lai uzturētu daudzvalodu vietni;
- Zend_Form- atvieglo formu veidošanu. Tas risina uzreiz vairākus uzdevumus, tādus kā:
- ievadīto elementu filtrēšana un validēšana;
- Elementu pārvirzīšana, glupēšana, sakārtošana;
- Zend_File_Transfer- atbalsta failu ielādēšanu, tas ietver datu validāciju un failu filtrus;
- Zend_Mail elektronisku vēstuļu sūtīšanai, var sūtīt vairākiem saņēmējiem;
- Zend_Measure- konvertors dažādiem lielumiem;
- Zend_Pdf- palīdz izveidot, jeb ielādēt failu .pdf formatā;
- Zend_ProgressBar- veicot kādu atjaunojumu, parādīsies logs ar laiku, cik vēl jāgaida, kamēr process pabeigsies.

Symphony

Symphony piedāvā veselu virkni ar spraudņiem (plugin). Tie ir apskatāmi pēc saites <http://www.symfony-project.org/plugins> . Viss ir sadalīts kategorijās, lietotājam ir iespējams izmantot meklēšanas sistēmu, lai atrastu sistēmai derīgo spraudni. Spraudņi ir izvēļjami atkarībā no Symphony versijas, objektu relāciju kartēšanas (*ORM*) un kategorijas.

- sfAdminThemejRollerPlugin- spraudnis, kas ļauj mainīt vietnes vizuālo noformējumu;
- sfDoctrineGraphvizPlugin- vizualizē datubāzes tabulu izskatu un saites;
- vjCommentPlugin- ļauj pievienot komentārus rakstiem;
- sfDoctrineJCroppablePlugin- palīdz apgriezt attēlus;
- sfMultipleAjaxUploadGalleryPlugin – vērtīgs spraudnis spēcīgas galerijas veidošanā;
- sfDoctrineForumPlugin- forums;
- psPageableFormPlugin- palīdz darbam ar vairākām lappusēm;
- sfGoogleLoginPlugin- pieslēgšanās sistēmai izmantojot google kontu;
- kdDoctrineGuardFacebookConnectPlugin- pieslēgšanās sistēmai izmantojot facebook kontu;
- drTwitterFeedPlugin- publicē izmaiņas lietotāja kontā Twitter vietnē;
- lbCartPlugin- grozs, pie tam glabā datus par izvēlētiem un nopirktam precēm, ļauj izmantot iespēju piedāvāt preces, kuras nopirkuši citi cilvēki, izvēlēties kādu preci no groza (cross-marketing). Ļoti vērtīga iespēja;
- tmcContactPlugin- sūta e-pastu, pēc formas aizpildīšanas;
- Šis ietvars piedāvā ļoti daudz interesantu spraudņu, daži no augstāk minētiem, nav vienīgie, kas dara aprakstītas darbības, eksistē dažādas iespējas vienas problēmas risinājumam.

CakePHP

Izstrādātāju komanda piedāvā lietotājiem sarakstu ar palīgiem:

- Form- palīgs darbam ar formu;
- Paginator palīgs darbam ar daudz lappušu vietni(katalogu);
- Rss- pierakstīšanās jaunumiem;
- Xml- palīdz veidot XML elementus.

Tie nav visi palīgi, bet sarakstā to nav daudz.

No vērtīgiem komponentiem ir Email, kas veido e-pasta ziņojumus un sūta tos.

Iespējama daudzvalodu vietņu veidošana ar Translate uzvedību.Kā arī Tree uzvedība koku veidošanai (kategorijas ar vairākām apakškategorijām).

Ļoti vērtīga arī ir iespēja dažāda veida failu ģenerēšanai. Dokumentācijā rakstīts, ka var ne vien noģenerēt kontrollerus, skatus, modeļus utt., bet arī funkcionāls būs pievienots automātiski, veidojot funkcionālo aplikāciju.

Saraksts ar interesantām iespējām, nav liels, bet grāmatā Cakebook[13], iespējams atrast labus problēmu risinājumus, piemēram, kā realizēt grozu.

Yii Framework

Tii Framework ietver sevī ļoti lielu paplašinājumu skaitu.

- yii-user- ļoti spēcīgs paplašinājums darbībām ar lietotājiem. Tas ietver sevī lietotāju reģistrāciju, autorizāciju, pieejas kontroli, ļauj lietotājiem veikt izmaiņas savā profilā, ļauj administratoram pārvaldīt lietotājus, iespējama kontu aktivācija caur e-pastu, parolu atjaunošana utt.
- Image- paver iespēju dažādā veidā rīkoties ar attēliem, mainīt lielumu, apgriezt, rotēt un darīt attēla krāsas asākas.
- Giix- koda ģenerators- ģenerē modeļus no datu bāzes, atvieglo darbu ar saitēm, lauku nosaukumu maiņu;
- Cfile- paplašinājums darbam ar failu sistēmu. Ļauj veikt ļoti daudz darbību, pārbauda faila/direktorijas esamību, lasamību, vai drīkst tajā ko ierakstīt, atgriež vārdu, tipu, paplašinājumu, nosaka izmēru utt.;
- Mbmenu un Cmenu- veido sarakstus;
- Fancybox-parāda attēlus nenoteiktajā secībā;
- Ejuidatetimepicker- ļauj izvēlēties datumu no kalendāra;
- Wsocialbutton- google+, twitter, facebook pogas ;
- Translate- modulis tulkošanai, tiek pievienota poga, kas ļauj izmantot google translate;
- Breadcrumbs- ļauj pievienot ceļu uz noteiktā tīmekļa lapām (piem., izvēloties kādu apakšskategoriju, augšā būs ceļš uz starta lapu un kategoriju sarakstu);
- Pdf- paplašinājums ļauj pārtaisīt HTML uz PDF formātu;
- Usercounter- lietotāju skaitītājs;
- Mprint- drukā to, ko lietotājs redz uz ekrāna;

- facebook-connect- pieslēdz lietotāju bez papildus tabulas datu bāzē, tikai ar facebook datiem.
- Paplašinājumu saraksts ir milzīgs, pie tam tas ir tikai oficiālajā *Yii* tīmekļa vietnē. Eksistē arī daudz citu pielikumu, piemēram, *Shopping Cart*- grozs, ļoti populārs pielikums interneta veikaliem.

Secinājumi

Par labāku ietvaru dažāda veida pieslēdzāmo moduļu, paplašinājumu, bibliotēku vērtīguma ziņā, viennozīmīgie līderi ir *Symfony* un *Yii Framework*. *Zend Framework* arī ir diezgan spēcīgs, bet oficiālie resursi to tik labi neparāda. *CakePHP* un *CodeIgniter* satur sevī vairākas vērtīgas papildiespējas, bet pārējie ietvari šajā ziņā ir daudz spēcīgāki. *Symfony* izceļas ar savu spraudņu atlasīšanas sistēmu, bet *Yii* vienkārši pārspēj visus paplašinājumu skaita ziņā. Pie tam *Yii* lietotāji ir aicināti novērtēt katru moduli, tādējādi lietotāji var redzēt, ko par šo paplašinājumu domā citi.

2.2.8. Vērtējumu tabula

Ietvaru salīdzināšanas rezultātā tika izveidota tabula ar iegūtiem rezultātiem (sk. 2.1. tab.). Atzīmes bija liktas piecu ballu sistēmā, kur pieci iegūst par labāko rezultātu. Balles tika liktas par ietvara uzstādīšanu, programmu izstrādāšanu, ātrumu, drošību, dokumentācijas skaidrojumiem un par citu vērtīgu pielikumu un moduļu esamību.

Ietvaru vērtējumu salīdzināšanas tabula

| | CodeIgniter | Zend Framework | Symfony | CakePHP | Yii Framework |
|--------------------------------|-------------|----------------|-----------|-----------|------------------|
| Uzstādīšana | 5 | 4 | 4 | 3 | 5 |
| Programmas izstrādāšana | 4 | 4 | 3 | 4 | 5 |
| Ātrums | 5 | 4 | 3 | 4 | 5 |
| Drošība | 4 | 4 | 4 | 4 | 5 |
| Dokumentācija | 5 | 4 | 5 | 3 | 5 |
| Citi | 3 | 4 | 5 | 3 | 5 |
| Kopā | 26 | 24 | 24 | 21 | <u>30</u> |

Secinājumi

Rezultātā tīmekļa vietnes izstrādāšanai tika izvēlēts *Yii Framework*. Izvēle tika bāzēta ne vien vērtējuma tabulas rezultātu dēļ, bet arī tāpēc, ka darba gaitā ietvars atstāja labu iespaidu. Ietvara iespējas un piedāvātie paplašinājumi der interneta veikala izstrādāšanai, padara darba procesu vieglāku, ātrāku un minimizē kļūdu skaitu.

Yii Framework piedāvā modeļu ģenerēšanu no datu bāzes, kā arī *CRUD* operāciju ar datu bāzi veikšanai nepieciešamo failu automātisko ģenerēšanu. Ļoti patika ietvara izstrādātāju produkta padeve, ieinstalējot ietvaru, tiek parādīts neliels tīmekļa vietnes piemērs. Pēc kura loģikas ir viegli un saprotami veidot turpmāko darbu. Dokumentācija piedāvā labus skaidrojumus, plašajām ietvara iespējām. Paplašinājumi piedāvā ļoti daudz dažādu prasību risinājumus.

Tīmekļa vietnes izstrādāšanas gaitas apraksts tiek aprakstīts nākamajā nodaļā.

4. INTERNETA VEIKALA IZSTRĀDĀŠANA

3.1. Apraksts

- Interneta veikala izstrādāšanai tika izvirzītas sekojošas prasības:
- Preču katalogs;
- meklēšanas sistēma;
- informatīva satura lapas;
- preču grozs;
- reģistrācija, autorizācija;
- auto rezerves daļu pasūtīšanas forma.

Interneta veikals tika sadalīts uz divām daļām lietotāja (*frontend*) un administratora (*backend*) daļām. Rezultātā arī izvirzītas prasības tika sadalītas un īstenotas attiecīgi tajā daļā, kur tām būtu jāstrādā. Veikala vizuālais noformējums tika veidots pēc agrāk minētas struktūras (skeleta), rezultātā lietotāji redz interneta veikalu tā, ka tas ir parādīts uz 3.1. attēla. Strādājot ar preču katalogu, administratori strādā ar vienkāršota dizaina lapu (sk.3.2. att.).

The screenshot shows the homepage of www.aparts.lv. At the top, there is a navigation bar with the website logo and several menu items: 'Uz sākumu', 'Īpašie piedāvājumi', 'Noformēt pasūtījumu', 'Palīdzība', 'Rakstiet mums', 'Autorizēties', and 'Reģistrēties'. Below the navigation bar, there is a search bar with a dropdown menu set to '-Visas kategorijas-' and a 'Meklēt' button. The main content area is divided into two columns. The left column contains a list of product categories with blue links: 'Aizdedzes sistēma', 'BERU', 'BOSCH', 'NGK', 'Akumulatori', 'DETA', 'Sznajde', 'Bosch', 'Autopiederumi', 'Degvielas kannas', 'Ugunsdzēsējamie aparāti', 'Auto spuldzes', 'NARVA', 'Philips', 'Osram', 'Autokosmētika', 'Bremžu sistēma', 'Brembo', 'TRW', 'Comline', 'PAGID', 'TRUSTING', 'Koivunen', 'FERODO', 'TEXTAR', 'ICER', 'Eļļas un smērvielas', and 'Castrol'. The right column contains a message: 'Mēs jau strādājam testa režīmā. Tagad ir iespējams sazināties ar mums izmantojot Skype (Nospiediet uz Skype pogas zem teksta). Šādi var uzzināt arī nepieciešamās preces cenu.' Below this message, there is a list of items for sale: 'Ja Skype atrodas režīmā "offline", rakstiet uz e-pastu apartsiv@inbox.lv. Neaizmirstiet uzrakstīt automašīnas gadu, dzinēja tilpumu, marku un modeli, vai arī detaļas kodu, jāuzraksta arī piegādes vieta. (Piemēram: BMW 320i, benzīns, automāts, 2002 gads, sedans- 1. priekšējie bremžu diski; 2. priekšējie bremžu kluči. Rīga, Pļavnieki)'. Below the list, there is a status indicator showing a green checkmark and the text 'I'm online'. At the bottom right, there is a small box with the text 'Preces: 0', 'Summa: 0 Ls.', and a link 'Apskatīt orozu'.

3.1. att. www.aparts.lv sākuma lapas izskats

Strādājot ar preču katalogu, administratori strādā ar vienkāršota dizaina lapu(sk. 3.2. att.). Pēc vajadzīgas kategorijas izvēlēšanas, labajā stūrī parādās vertikālā navigācija ar piedāvātajām iespējām.



3.2. att. www.aparts.lv administratora lapas izskats

3.2.Administratīvās daļas izstrādāšana

Administratora daļa ir domāta darbam ar preču katalogu. Administratoram ir iespēja strādāt ar kategorijām, apakškategorijām un precī. Šīs pozīcijas ir iespējams pievienot, dzēst un labot.

Kataloga pārvaldīšanas realizēšanas pamatā tika izmantota *Yii Framework* iespēja ģenerēt CRUD. Tā ir iespēja automātiski izmantojot modeli no datu bāzes noģenētēt kodu, kas ļautu izmantot biežāk lietojamas operācijas. Sākumā modulis izveido modeli (sk.3.3. att.), bet pēc tā izveidošanas tiek piedāvāts izmantot *CRUD* ģeneratoru (sk. 3.4. att.). Šādi administrātorā daļā parādās saraksts ar tabulu nosaukumiem (nosaukumi var būt mainīti), izvēloties kuru, tiek parādīts pozīciju saraksts un iespējamās darbības.

Model Generator

Model Generator

This generator generates a model class for the specified database table.

*Fields with * are required. Click on highlighted fields to edit them.*

Table Prefix
[empty]

Table Name *

Model Class *

Base Class *
CActiveRecord

Model Path *
application.models

Code Template *
default (D:\yii\framework\gii\generators\model\templates\default)

3.3. att. Modeļu ģeneratora izskats

- Generators
- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

Crud Generator

This generator generates a controller and views that implement CRUD operations for the specified data model.

*Fields with * are required. Click on the highlighted fields to edit them.*

Model Class *

Controller ID *

Base Controller Class *
Controller

Code Template *
default (/home/ardlv/public_html/yii/framework/gii/generators/crud/templates/default)

| Code File | Generate <input checked="" type="checkbox"/> |
|-------------------------------------------------|----------------------------------------------|
| controllers/itemsController.php | new <input checked="" type="checkbox"/> |
| views/items/_form.php | new <input checked="" type="checkbox"/> |
| views/items/_search.php | new <input checked="" type="checkbox"/> |
| views/items/_view.php | new <input checked="" type="checkbox"/> |
| views/items/admin.php | new <input checked="" type="checkbox"/> |
| views/items/create.php | new <input checked="" type="checkbox"/> |
| views/items/index.php | new <input checked="" type="checkbox"/> |
| views/items/update.php | new <input checked="" type="checkbox"/> |
| views/items/view.php | new <input checked="" type="checkbox"/> |

3.4. att. GRUD ģeneratora izskats

Uz 3.5. attēla ir redzams piemērs, kā tiek izvadīts apakškategoriju saraksts, kurā ir iespējams veikt meklēšanu un citas darbības.

Apakškategoriju saraksts

Var lietot salīdzināšanas operatorus (<, <=, >, >=, <> vai =) meklēšanas kritēriju sākumā, lai parādītu, kā salīdzināšanai ir jānotiek.

[Izvērstā meklēšana](#)

Displaying 1-10 of 21 result(s).

| id | Apakškategorijas nosaukums LV | Название подкатегории РУ | Kategorija | |
|----|-------------------------------|--------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | | |
| 4 | BERU | BERU | Aizdedzes sistēma |   |
| 5 | BOSCH | BOSCH | Aizdedzes sistēma |   |
| 6 | NGK | NGK | Aizdedzes sistēma |   |
| 7 | DETA | DETA | Akumulatori |   |

3.5. att. Noģenerētas lapas piemērs

Šāda veida automātiskā ģenerēšana paātrina izstrādes procesu. Tā ka, pārsvarā, visas nepieciešamas operācijas tika izdarītas, tās vajadzēja tikai pielabot. Piemēram modeļiem tika pievienotas relācijas. Lai, piemēram, veidojot, vai apskatot apakškategorijas, uz ekrāna izvadīt nevis piederīgas kategorijas ID, bet tās nosaukums. Preču pievienošanas formai tika pielikti divi atkarīgie izkrītošie saraksti ar kategorijām un apakškategorijām, kā arī iespēja ielādēt fotogrāfijas.

CRUD ģenerators ne vien piedāvā iespēju automatizēt koda veikšanu, bet rezultāta izveidotajām formām automātiski arī tiek veikta datu validācija, līdz ar to tikai datu bāzei derīgie dati iekļūs sistēmā.

3.3.Lietotāja daļas izstrādāšana

Lietotāja daļa sastāv no sekojošiem elementiem:

- Horizontālā navigācija;
- vertikālā navigācija;
- meklēšanas sistēma;
- preču grozs;
- kājene un galvene.

Zemāk tiek apskatītas katra elementa precīzas iespējas.

Horizontālā navigācija

Šī ir navigācija, kas ļauj pārvietoties pa vietnes sadaļām (sk.3.6. att.).

Uz sākumu Īpašie piedāvājumi Noformēt pasūtījumu Palīdzība Rakstiet mums Autorizēties Reģistrēties

3.6. att. Horizontālā navigācija

- 1.Uz sākumu – ir iespēja pārvietoties uz interneta sākulapu.
- 2.Īpašie piedāvājumi – piedāvā preču sarakstu ar atlaidēm.
- 3.Noformēt pasūtījumu - ļauj autorizētam lietotājam noformēt preču pasūtījumu.
- 4.Palīdzība – ir nodaļa, kura apkopota visa pircējam nepieciešama informācija, tāda ka: standartie piegādes nosacījumi, garantija, preces atgriešanas nosacījumi, rezerves daļu pasūtīšana.
- 5.Rakstiet mums – ir sadaļa ar kontaktformu, kura lietotājs var uzrakstīt radušos jautājumus, kā arī pasūtīt vajadzīgas rezerves daļas, jeb preces, kas nav katalogā.
- 6.Autorizēties – šī sadaļa ļauj lietotājam veikt autorizāciju.
- 7.Reģistrēties – jauna lietotāja reģistrācija.
- 8.Pēdējie divi punkti, tiek mainīti uz „Iziet no sistēmas”, kas lietotājs ir autorizējies.

Meklēšanas sistēma

Meklēšana tiek realizēta piedāvājot lietotājam izvēlēties kategoriju, kurā meklēt (sk. 3.7. att.). Atslēgas vārdi tiek meklēti preču tabulas dažādos laukos (nosaukumā, firmās, aprakstos). Ja kategorija netiek izvēlēta, preču atlasīšana notiek vīno visa kataloga, taču izvēloties konkrētu kategoriju, tiek meklēti tikai tai piederošas preces.



3.7. att. Meklēšanas sistēma

Vertikālā navigācija, jeb preču kategoriju saraksts

Tā kā tika nolemts, ka lietotāja daļas labajā pusē vienmēr ir jābūt preču kategoriju sarakstam, tika izmantota Yii piedāvāta iespēja izmantot lapu šablonus. Rezultātā par izskatu atbild šablons, kurā kategoriju saraksts paliek nemainīgs, bet centrālā daļa paliek visai informācija, kura pieejama vietnē.

Preču kategorijas tiek izvadītas uz ekrāna, kopā ar tām piederošām apakškategorijām. Izvēloties kategoriju, lapas centrā tika parādīti attēli, jeb apakškategoriju saraksts (sk.3.8. att.).



3.8. att. Sākuma lapas izskats

Izvēloties apakš kategoriju, tiek parādīts preču saraksts (sk 3.9. att.). Ja kategorijai nav apakš kategoriju, lietotājs uzreiz redz preču sarakstu.

Kopējā preču sarakstā tiek parādītas preču bildes kopā ar nosaukumiem un cenu, kura tiek izcelta sarkanajā krāsā, ja ir atlaide (sk. 3.9. att.).

Uz sākumu Īpašie piedāvājumi Noformēt pasūtījumu Palīdzība Rakstiet mums A

Home » Akumulatori » Bosch

Aizdedzes sistēma
BERU
BOSCH
NGK
Akumulatori
DETA
Sznajde
Bosch
Autopiederumi
Degvielas kannas
Ugunsdzēsāmie aparāti
Auto spuldzes

Bosch Silver Plus S4 001, 44Ah 39.00 Ls

Bosch Silver Plus S4 002, 52Ah 50.00 Ls

Bosch Hi-Tech AGM S6 001, 70Ah 105.00 Ls

3.9. att. Preču saraksts

Izvēloties konkrēto preci, lietotājs redz visu pieejamu informāciju, kā arī iespēju pievienot preci grozam (sk. 3.10. att.).

Uz sākumu Īpašie piedāvājumi Noformēt pasūtījumu Palīdzība Rakstiet mums

Home » Eļļas un smērvielas » Castrol

Aizdedzes sistēma
BERU
BOSCH
NGK
Akumulatori
DETA
Sznajde
Bosch
Autopiederumi
Degvielas kannas
Ugunsdzēsāmie aparāti
Auto spuldzes

NARVA
Philips
Osram

Castrol 10w40

Ražotājs: Castrol
Cena: 17.00 Ls

Grozā

Palielināt bildi

Preces apraksts: Magnatec Diesel 10W40 B4 4L

3.10. att. Preces apraksta izskats

Apskatīt interneta veikalu, kā arī pievienot preces grozā var visi lietotāji, taču noformēt preču pasūtījumu, iespējams tikai autorizētiem lietotājiem.

Reģistrācija/ autorizācija

Lai izmantotu lietotāju autorizāciju un reģistrāciju tika izmantots Yii User paplašinājums. Tas ir diezgan viegli pievienojams un pārveidojams pēc nepieciešamības. Lai paplašinājums atbilstu veikala prasībām, tika pievienoti papildus lauki par lietotājiem. Paplašinājums piedāvā divu veidu lietotāju piekļuvi: vienkāršie lietotāji un administratori.

Vienkāršie lietotāji var veikt izmaiņas tikai savā profilā, bet gan administratoriem pieejama visa cita informācija (izņemot paroli, tā ir šifrēta).

Reģistrācija veicama divos etapos. Sākumā lietotājs aizpilda formu(sk. 3.11. att.), bet pēc tam uz norādīto e-pastu tiek atsūtīts saite ar konta aktivizāciju.

- -

*Lauki atzīmēti ar * ir obligāti.*

Lietotājvārds *

Parole *

Minimāls paroles garums ir 4 simboli.

Atkārtot Paroli *

E Pasts *


Uzvārds *

Vārds *

Adrese (Valsts, pilsēta, iela, māja, dzīvoklis) *

Tālruna numurs *

Dzimšanas datums

Pārbaudes Kods *
 [Rādīt citus burtus](#)

Ievadiet burtus, kurus redzat attēlā.
Burti nav reģistra jūtīgi.

3.11. att. Reģistrācijas forma

Preču grozs

Preču grozs tika izveidots izmantojot paplašinājumu Yii Shopping Cart [27]. Tas ir komponents groza realizācijai. Grozs ir kontainers, kas glabā pozīciju kolekcijas un metodēm, kas ar to strādā. Groza stāvoklis starp pieprasījumiem glabājas lietotāja sesijā. Lai grozs strādātu, modelim, kas to izmantos, jāīsteno IECartPosition klasi, jāpievieno nepieciešamas metodes.

Šis paplašinājums arī piedāvā izmantot savu atlaižu sistēmu, kas ir balstīta uz procentuālajām atlaidēm, taču interneta veikalā tas notiek citādi.

Lietotājs var pievienot, dzēst, labot preču skaitu un redzēt cik kas maksā un kāda summa sanāk rezultātā (sk. 3.12. att.).



The screenshot shows a shopping cart interface. At the top, there is a navigation bar with links: "Noformēt pasūtījumu", "Palīdzība", "Rakstiet mums", "Autorizēties", and "Reģistrēties". Below the navigation bar, the title "Preču grozs" is displayed. The main content area shows a table with the following data:

| Preces nosaukums | Cena | Daudzums | Summa | Dzēst |
|-------------------------------|-----------|--------------------------------|-----------|---------------------------------------------------------------------------------------|
| Castrol 10w40 | 17.00 Ls. | <input type="text" value="2"/> | 34.00 Ls. |  |

Below the table, the text "Kopā: 34.00 Ls." is displayed. At the bottom right, there is a link "Noformēt pasūtījumu".

3.12. att. Preču groza izskats

Pēc tam autorizētais lietotājs var veikt pasūtījuma noformējumu. Tā rezultātā administratoram un lietotājam tiks noģenerēts un nosūtīts fails .pdf formatā (sk. 3.13. att.), kur tiks atainota pircēja kontaktinformācija un pasūtāmas preces.

Pasūtīto preču saraksts

| Personīga informācija | | | |
|--------------------------------|-----------|---------------------------------|-----------|
| Vārds uzvārds: | | Viktorija Šijenoka | |
| Adrese: | | Rīga Adresez iela 50-8, LV-1029 | |
| Tālruņa numurs: | | 12313231 | |
| Epasts: | | kizka@inbox.lv | |
| Groza saturs | | | |
| Preces nosaukums | Cena | Daudzums | Summa |
| Bosch Silver Plus S4 002, 52Ah | 50.00 Ls | 1 gab. | 50.00 Ls |
| Bosch Hi-Tech AGM S6 001, 70Ah | 105.00 Ls | 1 gab. | 105.00 Ls |
| Castrol 10w40 | 17.00 Ls | 1 gab. | 17.00 Ls |
| Summa kopā: 172.00 Ls | | | |

Paldies par pasūtījumu! Drīz ar Jums sazināšies un atsūtīs rēķinu.

Pasūtījums veikts: 05.02.12 18:17

3.13. att. Preču saraksts Pdf formātā

Pdf faila ģenerēšanai tika izmantots Yii piedāvātais paplašinājums TCPDF un pati klase. Paplašinājuma uzstādīšanas gaita tika aprakstīta Yii izstrādātāju vietē, bet pamācības, kā rīkoties ar klasi atrasti TCPDF vietnē [28, 29].

SECINĀJUMI UN PRIEKŠLIKUMI

Darba rezultātā tika sasniegts mērķis un izpildīti visi definētie uzdevumi.

Darba gaitā tika izpētīti pieci *PHP* ietvari, tādi ka: *CodeIgniter*, *Zend Framework*, *Symfony*, *CakePHP*, *Yii framework*. Tika analizētas tās teorētiskas iespējas, salīdzināti izstrādātāju apraksti, to dokumentācija. Teorētiskie dati un izstrādātāju apraksti tika pārbaudīti reālajā vidē, izmēģinot katra ietvara iespējas praktiski. Strādājot ar dažiem ietvariem, jau uzstādīšanas stadijā parādījās kādas problēmas, kuras bija risinātas, pirmkārt, izmantojot ietvara oficiālo dokumentāciju un, otrkārt, citus avotus. Pēc uzstādīšanas autores datorā, ar katru ietvaru tika izstrādātas divas programmas. Šādi tika iegūti rezultāti par ietvaru ātrumu, dokumentācijas pilnīgumu, uzstādīšanas, izstrādāšanas un izpratnes vieglumu, no jaunā ietvara lietotāja viedokļa. Tika apskatītas citas vērtīgas ietvaru īpašības un uz visu iegūtu rezultātu pamata tika izvēlēts ietvars interneta veikala izstrādāšanai. Ja darba sākumā par labāko ietvaru autorei likās *CodeIgniter* un *Zend Framework*, tad prakses gaitā un tās rezultātā tika izvēlēts *Yii Framework*.

Veicama darba beigās tika izstrādāts un aprakstīts auto piederumu interneta veikals, kas ir atbilstīgs izteiktām prasībām. Izstrādāšanai tika izmantots *Yii Framework* un tā paplašinājumi. Interneta veikalā ir pieejama lietotāju reģistrācija, autorizācija, preču pasūtīšana, groza izmantošana, pastāv iespēja sazināties ar pārdevēju un apskatīt preču grozu. Ir peejama informācija par preču pasūtīšanu, piegādi, garantiju un atgriešanas nosacījumiem. Veikalu ir iespējams pārvaldīt, labot kategoriju un apakškategoriju sarakstu, kā arī darboties ar precēm. Pēc preču pasūtīšanas uz klienta norādīto e-pastu tiek ģenerēts *pdf* fails ar pasūtīto preču sarakstu un cenām.

Pēc padarīta darba autore ieguva vērtīgu pieredzi un jaunas zināšanas par dažādiem *PHP* ietvariem. Kopumā ietvaru pētīšana un veikala izveidošana kļuva par patīkamu un kognitīvo procesu.

Pēc paveikta darba autore secināja, ka ietvara izmantošana vairākkārt atvieglo un paātrina šāda projekta izstrādāšanas procesu.

Lai izstrādāts interneta veikals būtu vēl labāks un pilnvērtīgāks, autores izteica sekojošus priekšlikumus:

- Pievienot sadaļu, kurā administrators varētu pievienot interneta veikala jaunumus un aktualitātes. Tām būtu automātiski jāpievienojas veikala sākumlapā;
- pievienot sadaļu, kura administrators varētu publicēt veikala tematikai atbilstošus rakstus;
- tā ka datu bāze ir gatava daudzvalodībai, īstenot vietnes satura pieejamību dažādās valodās.

IZMANTOTĀ LITERATŪRA

- 1., „Satura pārvaldības sistēma” Vikipēdijas raksts [tiešsaiste]. – [atsauce 2012.01.15.].
pieejams: http://lv.wikipedia.org/wiki/Satura_pārvaldības_sistēma.
2. Konceptija par elektronisku komerciju, MK akceptēta 2001. gada 13. martā.
- 3., „Elektroniskā komercija” LR Ekonomikas ministrija, 2007 [tiešsaiste]. – [atsauce 2011.01.15.].
pieejams: <http://www.em.gov.lv/e-komercija/page3a9f.html?id=6>
- 4., „*ab - Apache HTTP server benchmarking tool*” Apache Software Foundation raksts [tiešsaiste]. – [atsauce 2012.01.13.].
pieejams: <http://httpd.apache.org/docs/2.0/programs/ab.html>.
- 5., „CodeIgniter” Vikipēdijas raksts [tiešsaiste]. – [atsauce 2012.01.18.].
pieejams: <http://ru.wikipedia.org/wiki/CodeIgniter>.
- 6., „CodeIgniter User Guide Version 2.1.0” [tiešsaiste]. – [atsauce 2012.01.18.].
pieejams: http://codeigniter.com/user_guide/.
- 7., „Zend Framework” Vikipēdijas raksts [tiešsaiste]. – [atsauce 2012.01.20.].
pieejams: http://ru.wikipedia.org/wiki/Zend_Framework
- 8., „Zend Framework Overview” [tiešsaiste]. – [atsauce 2012.01.20.].
pieejams: <http://framework.zend.com/about/>
- 9., „Zend Framework Programmer's Reference Guide” [tiešsaiste]. – [atsauce 2012.01.21.].
pieejams: <http://framework.zend.com/manual/en/>.
10. Symfony izstrādātāju oficiālā vietne – [atsauce 2012.01.23.].
pieejams: <http://symfony.com>.
- 11., „A Gentle Introduction to symfony 1.4” online grāmata [tiešsaiste]. – [atsauce 2012.01.22.].
pieejams: http://www.symfony-project.org/gentle-introduction/1_4/en/.
- 12., „Requirements for running Symfony2” – Symfony dokumentācija [tiešsaiste]. – [atsauce 2012.01.23.].
pieejams: <http://symfony.com/doc/current/reference/requirements.html>.
- 13., „CakePHP Cookbook 2.x. Manual.” – CakePHP dokumentācija [tiešsaiste]. – [atsauce 2012.01.27.].
pieejams: <http://book.cakephp.org/2.0/en/>

- 14., „About Yii” – Yii framework izstrādātāju oficiāla vietne [tiešsaiste]. – [atsauce 2012.02.2.]. pieejams: <http://www.yiiframework.com/about/>.
- 15., „Features of Yii” – Yii framework izstrādātāju oficiāla vietne [tiešsaiste]. – [atsauce 2012.02.4.]. pieejams: <http://www.yiiframework.com/features/>.
16. „Installation” – Yii framework izstrādātāju oficiāla vietne, pamācība pēdējai stabilai versijai 1.1 [tiešsaiste]. – [atsauce 2012.02.4.]. pieejams: <http://www.yiiframework.com/doc/guide/1.1/en/quickstart.installation>
17. „Настройка Zend Framework” [tiešsaiste]. – [atsauce 2012.02.7.]. pieejams: http://habrahabr.ru/blogs/zend_framework/75535.
- 18., „The Book” – Symfony 2.0 Book , izveidota maijā 14, 2012 (Pdf) [tiešsaiste]. – [atsauce 2012.02.11.]. pieejams: http://symfony.com/pdf/Symfony_book_2.0.pdf?v=2.
19. CakePHP 1.3 Cookbook. Manual – CakePHP izstrādātāju oficiālā vietne [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://book.cakephp.org/1.3/view/907/Developing-with-CakePHP>.
- 20., „Zend Framework’s Programmer's Reference Guide” [tiešsaiste]. – [atsauce 2012.02.15.]. pieejams: <http://framework.zend.com/manual/en/learning.quickstart.create-project.html> manual.
- 21., „Doctrine” – izstrādātāju oficiālā vietne [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://docs.doctrine-project.org/projects/doctrine-orm/en/2.0.x/reference/tools.html#reverse-engineering>.
- 22., „The CookBook” – Symfony 2.0 CookBook , izveidota maijā 14, 2012 (Pdf) [tiešsaiste]. – [atsauce 2012.02.11.]. pieejams: http://symfony.com/pdf/Symfony_cookbook_2.0.pdf?v=2.
- 23., „Symfony Blog” Symfony izstrādātāju Blogs par izmaiņām un jaunumiem [tiešsaiste]. – [atsauce 2012.02.15.]. pieejams: <http://symfony.com/blog/a-week-of-symfony-280-7-13-may-2012>.
- 24., „CakePHP” – izstrādātāju oficiālā vietne [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://cakephp.org/> .
- 25., „Yii – User” Yii framework izstrādātāju paplašinājums [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://www.yiiframework.com/extension/yii-user> .

26. „Security” Yii framework izstrādātāju raksts par drošību [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://www.yiiframework.com/doc/guide/1.1/en/topics.security>.
27. „Shopping Cart” Yii Framework paplašinājums groza izveidošanai [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://yiiext.github.com/extensions/shopping-cart-component/>.
28. „tcpdf” Yii Framework izstrādātāju paplašinājums Pdf failu ģenerēšanai [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://www.yiiframework.com/extension/tcpdf#hh2>
29. „PHP class for generating PDF documents” TCPDF izstrādātāju oficiālā vietne [tiešsaiste]. – [atsauce 2012.02.13.]. pieejams: <http://www.tcpdf.org/>