

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**PHP IETVARU SALĪDZINĀJUMS UN IZVĒLE  
ATKARĪBĀ NO PROJEKTU MĒRĶIEM**  
BAKALaura DARBS

Autors: **Krišjānis Kokorīts**

Studenta apliecības Nr.: kk09303

Darba vadītājs: Dr. vadībzin. doc. Imants Gorbāns

RĪGA 2013

## ANOTĀCIJA

Šī darba mērķis ir apskatīt un rūpīgi salīdzināt pašlaik populārākos PHP ietvarus, kā arī novērtēt, cik piemēroti tie būtu dažādu projektu mērķiem. PHP valoda ir viena no pašlaik pasaulē populārākajām programmēšanas valodām. PHP ietvari ļauj izveidot spēcīgas tīmekļa lietotnes visdažādākajiem projektiem. Salīdzinot ietvarus, ņemti vērā tādi būtiski faktori kā ietvara uzstādīšanas un jauna projekta izveides vieglums, konfigurācijas pārvaldība, ievadformu datu pārbaude un ģenerēšana, datubāzes pārvaldība, pieejamās ietvaru bibliotēkas u.c. faktori. Projekta mērķu ziņā ņemti vērā tādi faktori kā projektējamās sistēmas lielums, sistēmas nākotnes perspektīvas, projektējamās sistēmas kvalitāte un ātrdarbība u.c.

Atslēgvārdi: PHP, ietvari , projekti, MVC, CakePHP, CodeIgniter, Yii, Symfony.

## **ABSTRACT**

The aim of this work is to look carefully and compare the currently most popular PHP frameworks, as well as to appraise their suitability to different objectives of the projects. PHP language currently is one of the world's most popular programming languages. PHP framework enables you to create powerful web applications in a wide range of projects. Comparison took into account such critical factors as framework installation and ease of creating new project, configuration management, entry form generation and form data validation, database management, available libraries etc. factors. Project objectives were taken into account in terms of such factors - size of system, the perspective of system's future, quality and speed of system etc.

Keywords: PHP, framework, projects, MVC, CakePHP, CodeIgniter, Yii, Symfony.

# SATURS

Definīcijas, saīsinājumi un apzīmējumi .....	6
Ievads .....	7
1. PHP programmēšanas valoda .....	8
2. Tīmekļa Lietotņu ietvari .....	9
2.1. Tīmekļa ietvaru raksturiezīmes .....	9
2.2. Tīmekļa ietvaru iedalījums .....	10
3. Populārākie PHP ietvari .....	12
4. PHP ietvaru arhitektūra .....	14
5. PHP ietvaru salīdzinājuma kritēriji.....	15
6. CakePHP.....	16
6.1. Tehniskās prasības .....	16
6.2. Uzstādīšana un jauna projekta izveide.....	16
6.3. Konfigurācijas pārvaldība.....	16
6.4. Datu bāzu pārvaldība un drošība .....	17
6.5. Ietvara iekšējā arhitektūra.....	17
6.6. Ievadformas, to datu pārbaude un ģenerēšana.....	18
6.7. Speciālas iespējas .....	19
6.8. Trūkumi .....	20
7. CodeIgniter .....	21
7.1. Tehniskās prasības .....	21
7.2. Uzstādīšana un jauna projekta izveide.....	21
7.3. Konfigurācijas pārvaldība.....	21
7.4. Datu bāzu pārvaldība un drošība .....	22
7.5. Ietvara iekšējā arhitektūra.....	23
7.6. Ievadformas, to datu pārbaude un ģenerēšana.....	24
7.7. Speciālas iespējas .....	25
7.8. Trūkumi .....	26

8. Symfony .....	27
8.1. Tehniskās prasības .....	27
8.2. Uzstādīšana un jauna projekta izveide.....	27
8.3. Konfigurācijas pārvaldība.....	27
8.4. Datu bāzu pārvaldība un drošība .....	28
8.5. Ietvara iekšējā arhitektūra.....	28
8.6. Ievadformas, to datu pārbaude un ģenerēšana.....	29
8.7. Speciālas iespējas .....	30
8.8. Trūkumi .....	31
9. Yii.....	32
9.1. Tehniskās prasības .....	32
9.2. Uzstādīšana un jauna projekta izveide.....	32
9.3. Konfigurācijas pārvaldība.....	32
9.4. Datu bāzu pārvaldība un drošība .....	33
9.5. Ietvara iekšējā arhitektūra.....	34
9.6. Ievadformas, to datu pārbaude un ģenerēšana.....	35
9.7. Speciālas iespējas .....	35
9.8. Trūkumi .....	35
10. Projekti un tiem atbilstošu ietvaru izvēle .....	36
10.1. Teorētisks projektu apskats .....	36
10.2. Praktisks projektu apskats .....	38
Rezultāti .....	41
Secinājumi.....	42
Izmantotās literatūras saraksts.....	43
Pielikumi .....	45

## DEFINĪCIJAS, SAĪSINĀJUMI UN APZĪMĒJUMI

**MVC** – strukturēšanas princips programminženierijā.

**HTTP** – hiperteksta transporta protokols, kas galvenokārt tiek izmantots tīmekļu vietnēs.

**ORM** – objektu relāciju kartēšana.

**SQL** – strukturēta vaicājumvaloda.

**CRUD** – četras pamata funkcijas : pievienot, atgūt, modificēt, izmest.

**XML** – paplašināmās iezīmēšanas valoda.

**JSON** – Javascript objektu notācija.

**RSS** –satura vienkāršā sindicēšana.

**CSV**– ar komatu atdalītu vērtību failu formāts.

**GPL** – bezmaksas programmatūras licence.

**URL** – vienotais resursu vietrādis.

**LaTeX** – dokumentu atzīmēšanas valoda.

**CMS** – satura vadības sistēma. Izmanto, lai pārvaldītu tīmekļa vietņu dinamisko saturu.

## IEVADS

Mūsdienās eksistē daudzi un dažādi PHP ietvari. Ir grūti starp daudzajiem ietvariem izvēlēties piemērotāko konkrētā projekta mērķiem. Bakalaura darba mērķis ir salīdzināt populārus PHP ietvarus un noskaidrot, kā tie der dažādiem projektu uzdevumiem. Lai to izdarītu, ir sastādīti kritēriji, pēc kā novērtēt ietvarus. Izvēloties kritērijus, ņemts vērā, lai kritēriji palīdzētu programmētājiem izvēlēties labāko ietvaru projekta mērķiem, un lai lietotne, kas tiktu izstrādāta konkrētajā ietvarā, varētu papildīt klientu vēlmes un cerības. Veikts pētījums, salīdzinot ietvarus gan, teorētiski iepazīstoties ar pieejamo informāciju par ietvariem, gan, praktiski izstrādājot tīmekļa lietotnes ietvaros. Darba tēmas izvēli nosaka praktiska nepieciešamība uzzināt, kādā gadījumā kādus ietvarus vislabāk izvēlēties.

Darba struktūra: sākumā apskatīta teorija par PHP programmēšanas valodu, ietvariem, to arhitektūru un definēti kritēriji, pēc kuriem salīdzināt ietvarus. Pēc tam salīdzināti pašlaik populārākie ietvari – CodeIgniter, CakePHP, Symfony, Yii pēc salīdzināšanas kritērijiem. Darba nobeigumā izvērtēti kādiem projektiem šie ietvari vislabāk der.

# 1. PHP PROGRAMMĒŠANAS VALODA

PHP ir servera puses skripta valoda, domāta tīmekļa izstrādei. PHP pašlaik tiek lietota vairāk nekā 244 miljonos saitu un ir uzinstalēta uz apmēram 2,1 miljona tīmekļa serveru. Tā ir viena no populārākajām tīmekļa programmēšanas valodām visā pasaulē. Oriģināli PHP valodu izveidojis Rasmus Lērdorfs (*Rasmus Lerdorf*) [16], un saīsinājums PHP nozīmē „*Personal Home Page*” [1]. Tagad to atšifrē kā „*PHP: Hypertext Preprocessor*”. Serveris PHP kodu interpretē ar PHP procesora moduli, ar kuru tas ģenerē attēlojamo tīmekļa lapu. PHP komandas var tikt pa tiešo liktas HTML dokumentā, nevis tikt izsauktas kā atsevišķi faili, lai apstrādātu datus. Jaunākajām PHP versijām ir pieejama arī komandrindas saskarnes iespēja, un PHP valoda var tikt lietota arī savrupās (*standalone*) grafiskās lietotnēs. PHP ir bezmaksas programmatūra, kas izlaista ar PHP licenci, kas atšķiras no GNU licences *General Public License* (GPL) vienīgi ar aizliegumu lietot terminu PHP. PHP programmatūra var tikt instalēta uz vairuma tīmekļa serveru, uz gandrīz visām operētājsistēmām un platformām. PHP programmēšanas valoda ir speciāli taisīta līdzīga C programmēšanas valodai, kas arī izskaidro šīs valodas lielo popularitāti visā pasaulē.

## 2. TĪMEKĻA LIETOTŅU IETVARI

Tīmekļa izstrāde ir plašs jēdziens, kas apzīmē darbu, kas vajadzīgs, lai izstrādātu tīmekļa lapas, lietotnes internetam vai intranetam. Tīmekļa lietotņu ietvari ir programmatūras ietvari, kas tiek veidoti, lai atbalstītu dinamisku tīmekļa lapu, tīmekļa lietotņu un tīmekļa servisu izstrādi. Ietvaru mērķis ir atvieglot programmētājiem darbu, izstrādājot dažādas tīmekļa lietotnes, piemēram daudzi ietvari piedāvā bibliotēkas, lai varētu pieslēgties datubāzēm, veidot sesijas, veidot autentifikācijas sistēmas utt. Ietvari bieži vien piedāvā atkārtoti izmantot kodu un palīdz labi strukturēt projektu.

### 2.1. Tīmekļa ietvaru raksturiezīmes

Ietvariem piemīt šādas raksturiezīmes:

- *WEB template system* – tīmekļa veidnes sistēma, kad tīmekļa lapas sastāv no statiskas daļas (HTML) un dinamiskas daļas – koda, kas ģenerē HTML.
- *WEB caching* – īslaicīga tīmekļa lapu glabāšana, lai palielinātu lapas ielādes ātrumu, servera ātrumu.
- *Security* – drošība. Daži ietvari nāk kopā ar autentifikācijas un autorizācijas mehānismiem.
- *Scaffolding* – daži ietvari ģenerē kodu, kas ļauj lietotnei pievienot, lasīt, modificēt, izmest (CRUD) datubāzes ierakstus.
- *URL mapping* – tīmekļa adreses veidošanas mehānisms, ko veic ietvars.
- *Ajax* – tīmekļa izstrādes tehnika, kas ļauj taisīt interaktīvas tīmekļa lietotnes.
- *Web services* – daži ietvari piedāvā rīkus, lai taisītu un uzturētu tīmekļa servissus.

## 2.2. Tīmekļa ietvaru iedalījums

Ietvarus var iedalīt pēc vairākiem kritērijiem.

### 2.2.1. Iedalījums pēc strukturēšanas principa

Visvairāk tīmekļa lietotņu ietvaru ir balstīti uz MVC strukturēšanas principa, taču ir arī citi strukturēšanas principi.

#### **Strukturēšanas principi:**

- ***trīsriindu arhitektūra (three –tier organization)***

Šajā arhitektūrā lietotne tiek strukturēta ap trīs fiziskajiem līmeņiem – klientu, lietotni, datubāzi. Tīmekļa lietotnes ar klienta līmeni saprot tīmekļa pārlūku, bet ar lietotnes līmeni saprot lietotnes galveno loģiku. Šādas arhitektūras piemērs ir #C programmēšanas valoda .NET platformā.

- ***satura vadības sistēmas (Content Management System)***

Satura vadības sistēmas ir sistēmas, kas ļauj publicēt, rediģēt to saturu, kā arī ļauj rediģēt saita struktūru no galvenās lapas. Šādas sistēmas ir WordPress, Joomla , Drupal u.c.

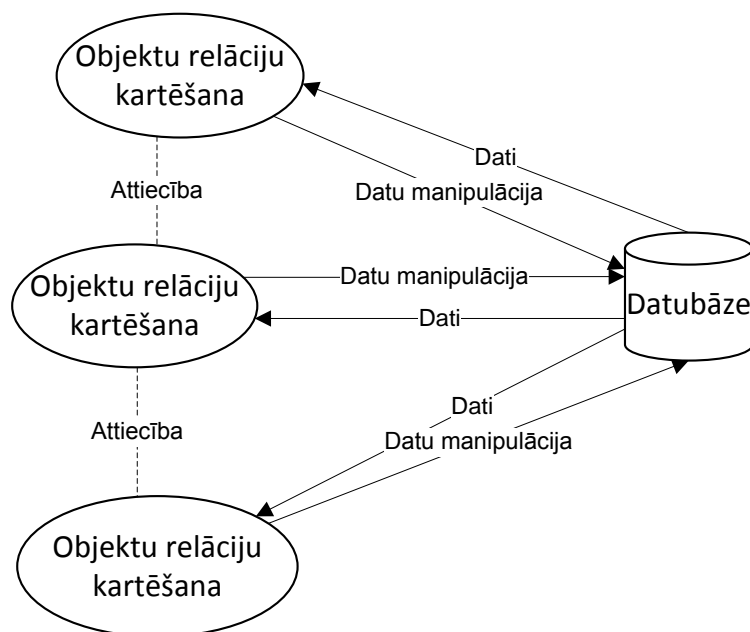
- ***Model View Controller***

MVC strukturēšanas princips ir vairāk aprakstīts 4. nodaļā.

### 2.2.2. Iedalījums pēc tā, kā tiek veikta datu apmaiņa ar datubāzi

Ietvari izmanto šādas datu apmaiņas iespējas ar datubāzi:

- ***ORM (Object-relational mapping)*** – objektu relāciju kartēšana nodrošina savstarpēju datubāzes objektu saistību programmā. Objektu relāciju kartēšana parasti saista dažādas datubāzes tabulas, par pamatu izmantojot to ārējās atslēgas. Izmantojot objektu relāciju kartēšanu, var ērti saistīt biznesa loģikas elementus ar datubāzes elementiem, atvieglojot saistīto objektu ielādi un nodrošinot datu integritāti, ierakstu labošanas un dzēšanas gadījumā.



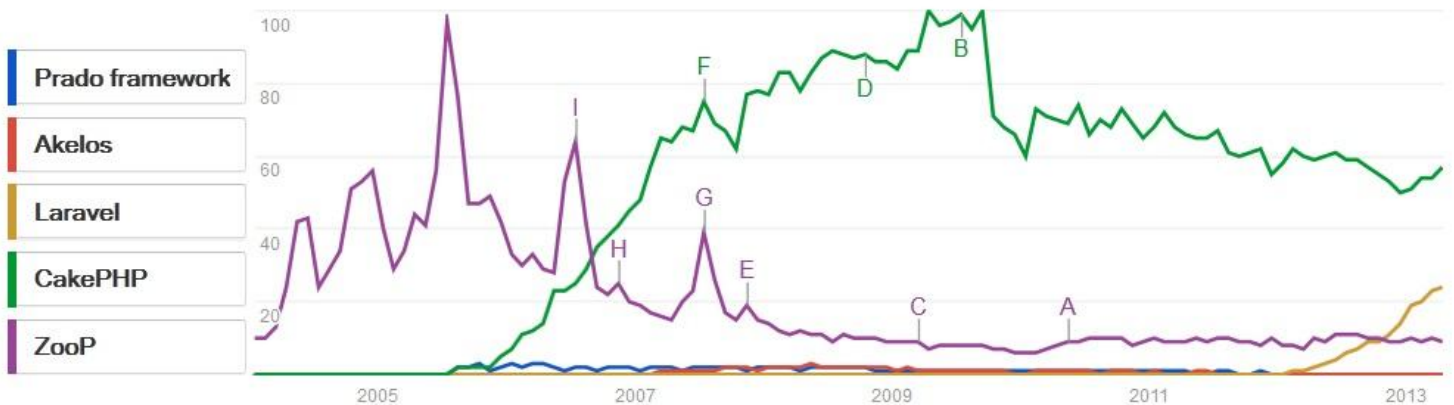
2.2.2.att., **ORM**

- **Query Builders** – vaicājumu būvētāji. Ja ietvari neizmanto ORM, tad bieži vien tie uzbūvē paši savus vaicājumu būvētājus, lai aizsargātu vaicājumus no SQL injekciju uzbrukumiem un atvieglotu programmētājiem vaicājumu rakstīšanu.

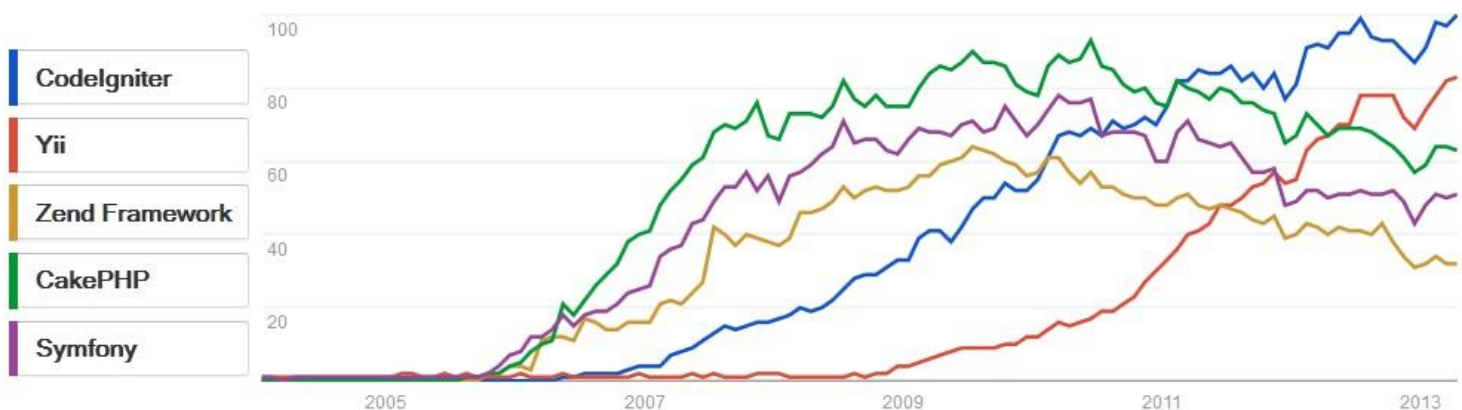
### 3. POPULĀRĀKIE PHP IETVARI

Lai noskaidrotu populārākos PHP ietvarus, tika lietots Google Trends [2]. Tas parāda Google tīmekļa meklētājprogrammas meklēto vienību skaitu noteiktā laika periodā. Diagrammā attēlo meklēšanas vaicājumu skaitu normalizētā skalā no 0 līdz 100. Par 100 tiek pieņemts augstākais punkts diagrammā (visvairāk meklējumi). Par 0 diagramma attēlo punktus, kur nav pietiekams apjoms meklējumu vai arī salīdzinoši ļoti maz meklējumu, ņemot vērā augstāko punktu diagrammā. Tā kā *Google Trends* ļauj ierakstīt tikai līdz piecām meklēšanas frāzēm, taču populāri ir vairāk par 5 ietvariem, tad, lai varētu salīdzināt, par pamatu tika izvēlēts CakePHP, kurš ir atlasīts visos meklējumos. Iegūtie rezultāti ir redzami 1., 2. un 3. attēlā.

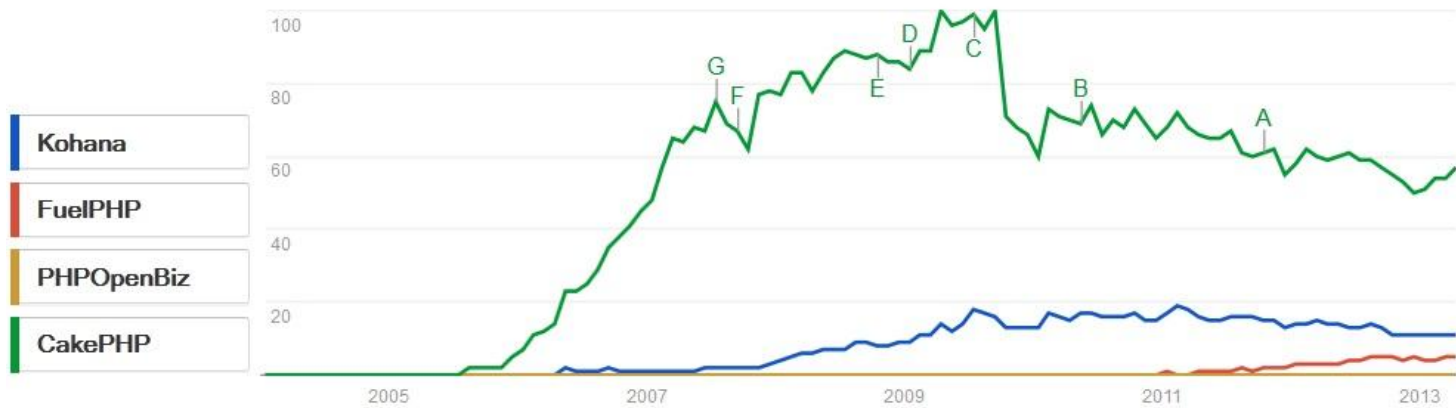
Šie dati ir nolasīti 2013.gada 1.maijā.



3.1. att., Google Trends - salīdzinājums -1



3.2. att., Google Trends - salīdzinājums -2



3.3. att., Google Trends - salīdzinājums -3

### Secinājumi par populārākajiem ietvariem

Apskatot grafikus un iepazīstoties ar aptaujām internetā [3] [4] var pateikt, ka pašlaik vispopulārākie ietvari ir šādi (sakārtoti, sākot ar vispopulārāko, ņemot vērā „Google Trends” rezultātus) :

- 1) CodeIgniter – pamazām, taču noturīgi tā popularitāte ir pieaugusi kopš 2006. gada, un tagad ir vispopulārākais ietvars google.com meklējumu ziņā,
- 2) Yii – popularitāti ieguvis diezgan pakāpeniski kopš 2009.gada,
- 3) CakePHP – viens no pirmajiem populārajiem ietvariem, kas popularitāti sāka iegūt 2005.gada nogalē,
- 4) Symfony – arī viens no pirmajiem populārajiem ietvariem, kas popularitāti sāka iegūt 2005.gada nogalē.

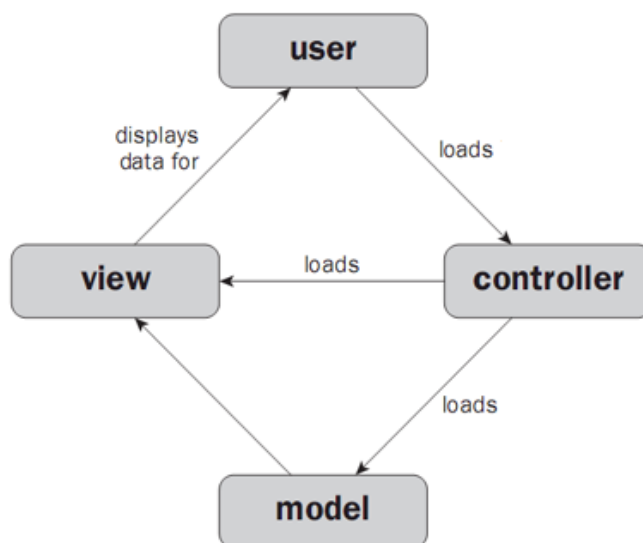
## 4. PHP IETVARU ARHITEKTŪRA

Vairums PHP ietvaru ir būvēti, izmantojot MVC strukturēšanas principu. MVC ir fundamentāls strukturēšanas princips, kas tiek lietots interaktīvās sistēmās [5][17]. MVC strukturēšanas principa galvenā doma ir atdalīt sistēmas prezentāciju no sistēmas loģikas. MVC ir vispopulārākais tīmekļa lietotņu ietvaru izmantotais strukturēšanas princips. Tam ir savs iemesls – labi sadalīta struktūra ļauj daudz vieglāk uzturēt un mainīt sistēmu.

Model -View- Controller strukturēšana sastāv no:

- Model (modelis) atbild par pieslēgšanos datubāzei, datu iegūšanu, saglabāšanu tajā (piem., php valodā),
- View (skats) atbild par datu attēlošanu lietotājam (piem., HTML, css, javascript valodās),
- Controller (kontrolieris) atbild par modeļu izsaukšanu, skatu ielādi, no skatiem izsaukto funkciju apstrādāšanu (piem., php valodā) .

MVC strukturēšanas principus apraksta zemāk esošais attēls.



4.. att., MVC uzbūve

Shēmā attēlots, kā var notikt izsaukumi. Lietotājs var izsaukt kontrolieri. Pēc tam kontrolieris var izsaukt modeļus un skatus. Modeļi var pieslēgties datubāzei un apstrādāt/iegūt datus, tad modelis var ielādēt datus skatā. Šī struktūra tiešām atvieglo dzīvi sistēmas uzturētājiem, jo veicina koda lasāmību.

## 5. PHP IETVARU SALĪDZINĀJUMA KRITĒRIJI

Darbā autors apskatīs un salīdzinās ietvarus pēc šādiem kritērijiem:

- **Tehniskās prasības**

Apraksta to, uz kādām PHP versijām iet ietvars. Ietvaram būtu jādarbojas uz praktiski visiem standarta hostinga serveriem, kuriem iespējots PHP5 un MySQL5.

- **Ietvara iekšējā arhitektūra**

Apraksta to, kāda ir PHP ietvara arhitektūra. MVC gadījumā apraksta, ko konkrētais PHP ietvars saprot ar skatiem, kontrolieriem un modeļiem.

- **Uzstādīšana un jauna projekta izveide**

Apraksta to, cik viegli var uzstādīt ietvaru un izveidot jaunu projektu, un, kas būtu jāzina par labāko veidu, kā to izdarīt.

- **Konfigurācijas pārvaldība**

Apraksta to, kāda ir PHP ietvara folderu struktūra un kur var konfigurēt tādas svarīgas lietas kā datubāzes pieslēgšanās datus u.c.

- **Datu bāzu pārvaldība un drošība**

Apraksta to, kādā veidā notiek interakcija ar datubāzi un vai un kādas ir pieejamās metodes datubāzes drošības nodrošināšanai (cīņai ar SQL injekcijām u.c.).

- **Ievadformas, to datu pārbaude un ģenerēšana**

Apraksta to, vai un kādus līdzekļus ietvars piedāvā datu ievadformu pārbaudei un ģenerēšanai.

- **Speciālas iespējas**

Apraksta to, kādas speciālās iespējas ir pieejamas, lietojot šo ietvaru, piemēram, palīgbibliotēkas u. taml.

- **Trūkumi**

Apraksta to, kādi trūkumi, nepilnības ir ietvaram.

## 6. CAKEPHP

PHP ietvars CakePHP zemāk tiek salīdzināts pēc salīdzināšanas kritērijiem.

### 6.1. Tehniskās prasības

CakePHP atbalsta gan PHP4, gan PHP5 versijas.

### 6.2. Uzstādīšana un jauna projekta izveide

CakePHP uzstādīšana ir samērā ātri paveicama un viegla. CakePHP mājaslapā ir labi aprakstīts, kas ir jādara, lai uzstādītu CakePHP [6]. Kad tas ir izdarīts, var ķerties klāt jau kontrolieru, modeļu un skatu veidošanai. CakePHP arī ir ieteicams iespējot tīmekļa adreses pārrakstīšanas moduli (*apache* servera gadījumā modificē `httpd.conf` failu).

### 6.3. Konfigurācijas pārvaldība

CakePHP 2.1.3. konfigurācijas pārvaldība tiek organizēta ar samērā intuitīvi saprotamu struktūru. Zemāk ir attēlota šī folderu struktūra. Tiek izcelti tie folderi un faili, kuru modificēšana ir vajadzīga jauna projekta izveidei.

- cakephp
  - ✓ app
    - ✓ **Config**
      - ✓ **database.php** - šeit jākonfigurē datubāzes pieslēgšanās dati
      - ✓ **routes.php** – šeit var uzstādīt noklusēto kontrolieri
    - ✓ Console
    - ✓ **Controller** - šeit atrodas kontrolieru php faili
    - ✓ **Lib** - šeit var likt savas izveidotās bibliotēkas
    - ✓ Locale
    - ✓ **Model** – šeit atrodas modeļu php faili
    - ✓ Plugin
    - ✓ Test
    - ✓ Tmp
    - ✓ **Vendor** – šeit var likt citu izveidotās bibliotēkas
    - ✓ **View** - šeit atrodas prezentēšanai domātie faili
      - ✓ Elements
      - ✓ Emails
      - ✓ Errors
      - ✓ Helper
      - ✓ Layouts
      - ✓ **Pages** - šeit jāglabā lapu faili
      - ✓ Scaffolds
  - ✓ Webroot

- ✓ **css** - šeit ieteicams likt savus css failus
- ✓ **files** – šeit ieteicams likt failus
- ✓ **img** – šeit ieteicams likt bildes
- ✓ **js** – šeit ieteicams likt savus javascript failus
- ✓ lib
- ✓ plugins
- ✓ vendors
- .htaccess

### 6.3. att., *CakePHP fiziskā struktūra*

## 6.4. Datu bāzu pārvaldība un drošība

CakePHP atbalsta pieslēgšanos pie vairāk kā vienas datubāzes vienlaicīgi, ko var ērti realizēt. CakePHP izmanto ORM. Lai datubāze tiktu pasargāta no SQL injekcijām, var lietot iebūvētās CakePHP SQL pieprasījumu ORM metodes - find() un save() ar piemērotu masīvu notāciju, piem. array('field' => \$value). ORM find() metodes piemērs ir attēlos zemāk.

```
$specificallyThisOne = $this->Article->find('first', array('conditions' => array('Article.id' => 1)));
```

### 6.4. att., *CakePHP ORM*

## 6.5. Ietvara iekšējā arhitektūra

CakePHP arhitektūra ir balstīta uz MVC strukturēšanas principiem, tāpēc CakePHP struktūra sastāv no skatiem, modeļiem un kontrolieriem. CakePHP modeļu lietošana ir obligāta, jo CakePHP lieto ORM. Modeļi tiek saprasti kā datu reprezentētāji un tiek lietoti, lai piekļūtu datubāzes tabulām. Katram modelim automātiski tiek definēta un atrasta tabula datubāzē ar tādu pašu datubāzes tabulas nosaukumu, kāds ir modelim. Modeļu un tabulu nosaukumiem eksistē speciāli noteikumi. Modeļi tiek saukti vienskaitlī, un tiem jāsākas ar lielo burtu. Ja modeļa nosaukums sastāv no vairākiem vārdiem, tad katram vārdam ir jāsākas ar lielo burtu. Tabulu nosaukumiem datubāzē ir jābūt daudzskaitlī, rakstītiem ar mazajiem burtiem. Ja tabula sastāv no vairākiem vārdiem, tad vārdiem ir jābūt atdalītiem ar „\_”. Ja definē modeli „RegisteredUser”, automātiski šim modelim tiks piesaistīta tabula „registered\_users”. Vajadzības gadījumā katram modelim var speciāli norādīt tabulu, kuru piesaistīt modelim, ja tabulu un modeļu nosaukumi atšķiras. Parasti CakePHP modelis sastāv no maz php rindiņām, jo, lai modelis veiksmīgi darbotos, ir nepieciešama tikai modeļa deklarācija, kā redzams zemāk esošajā attēlā.

```
RegisteredUser.php
1
2 <?php
3
4 class RegisteredUser extends AppModel
5 {   var $name = 'RegisteredUser';
6
7 }
8 >>
```

6.5.1.att., *Modeļa piemērs*

CakePHP ar skatiem saprot HTML, XML, JSON, PDF u.c. tipu failus.

Pēc MVC principa kontrolieriem ir biznesa loģikas funkcijas – skatu ielāde un datu apstrāde, izmantojot modeļus, tāpēc no CakePHP kontrolieriem tiek izsaukti skati un modeļi. Modeļu izsaukšana nozīmē datu iegūšanu, saglabāšanu modelim piesaistītajā tabulā, ko var vienkārši realizēt. Piemērā ir parādīta funkcija, kas izsauc skatu un modeli. Šī funkcija atrodas kontrolierī.

```
public function load_page($isn)
{
    $specificallyThisOne = $this->RegisteredUser->find('first', array('conditions' => array('isn' => $isn)));
    $this->set(compact('specificallyThisOne'));
    $this->render('/Pages/parameters');
}
```

6.5.2.att., *Skatu un modeļu izsaukšana*

## 6.6. Ievadformas, to datu pārbaude un ģenerēšana

Ievadformu datu pārbaudi var ērti realizēt ar CakePHP iebūvēto ORM datu saglabāšanas komandu „save”. Ievadformu validācijas likumus var pievienot vajadzīgajā modelī ar \$validate masīva palīdzību, kur atbilstošajiem ievadlaukiem pretī ir atbilstošie validācijas likumi, kas jau ir iepriekš definēti, kā ir redzams zemāk esošajā attēlā.

```
1 <php
2 class RegisteredUser extends AppModel {
3     var $name = 'RegisteredUser';
4     var $validate = array(
5         'login' => 'alphaNumeric',
6         'email' => 'email',
7         'born' => 'date'
8     );
9 }
```

6.6.1.att., *Validācijas likumu veidošana*

Validācijas likumus var samērā ērti un saprotami definēt.

## 6.7. Speciālas iespējas

CakePHP piedāvā iebūvētas palīgbibliotēkas, sistēmas komponentes un palīgus, kas var lieti noderēt, ja ir jāveic kādi ierasti programmēšanas uzdevumi.

Ir pieejamas šādas galvenās sistēmas **komponentes**:

Acl	- noder 'access control lists' veidošanai
Auth	- noder autentifikācijas veidošanai
Cookie	- darbojas līdzīgi 'Session' bibliotēkai, lieto PHP cookies
Email	- ļauj sūtīt e-pastus, lietojot Mail, SMTP u.c.
RequestHandler	- noder, lai iegūtu informāciju par saita apmeklētājiem
Security	- noder, lai uzstādītu papildus drošības pasākumus, darbojoties ar HTTP autentifikāciju
Session	- noder sesiju veidošanā, lieto PHP sesijas

Ir pieejams šādas **palīgbibliotēkas**:

App	- noder, lai ielādētu, izsauktu modeļus, komponentes, klases no modeļiem, skatiem
Inflector	- noder, lai manipulētu ar string tipa mainīgajiem, piem., atgriež string daudzskaitlī.
String	- noder, lai manipulētu ar string tipa mainīgajiem
Xml	- noder, lai veidotu, darbotos ar XML dokumentiem.
Set	- noder, lai darbotos, optimizētu darbu ar masīviem
Security	- noder, lai uzstādītu drošības pasākumus, piemēram, kriptogrāfijas funkcijas
Cache	- noder, lai darbotos ar kešatmiņas sistēmām
HttpSocket	- noder, lai darbotos ar pieprasījumiem, piem., tīmekļa servisu pieprasījumiem
Router	- palīdz darboties ar tīmekļa adreses sadalīšanas un veidošanas uzdevumiem

Bez tam CakePHP piedāvā arī **palīgus**( helpers) dažādu uzdevumu veikšanai, kuri labi noder skatos. Šie palīgi ir Ajax, Cache, Form, HTML, Js, Javascript, Number, Paginator, RSS, Session, Text, Time, XML.

## **6.8.Trūkumi**

CakePHP daudzas lietas notiek automātiski pēc konvencijām, kas zināmā mērā palēnina izpildes laiku. Iesācējiem varētu būt daudz jāmācās , lai saprastu CakePHP darbības principus un visas CakePHP daudzās konvencijas.

## 7. CODEIGNITER

PHP ietvars CodeIgniter zemāk tiek salīdzināts pēc salīdzināšanas kritērijiem.

### 7.1. Tehniskās prasības

CodeIgniter atbalsta gan PHP 5.1 un jaunākas versijas.

### 7.2. Uztādīšana un jauna projekta izveide

CodeIgniter uztādīšana ir samērā ātra, viegli saprotama un parocīga. CodeIgniter mājaslapā ir atrodama informācija [7] , kas jā dara, lai izveidotu jaunu projektu.

No lielākas drošības viedokļa ir ieteicams, ka CodeIgniter „system” un „application” folderi atrastos virs servera „root” direktorijas, lai tie būtu pilnībā nesasniedzami lietotājam, patvaļīgi mainot tīmekļa adresi. Lai gan CodeIgniter ir parūpējies, lai visos folderos būtu definēti .htaccess faili, kas aizliedz tiešu piekļuvi sistēmas failiem.

### 7.3. Konfigurācijas pārvaldība

CodeIgniter 2.1.0. konfigurācijas pārvaldība tiek organizēta ar samērā viegli atšifrējamu folderu struktūru, kas gan nedaudz atšķiras no iepriekšējās CodeIgniter 1.7.2 versijas, kas lietota iepriekš.

Zemāk ir attēlota šī folderu struktūra. Ir izcelti folderi un faili, kuru modificēšana ir vajadzīga jauna projekta izveidei. CodeIgniter\_2.1.0

- ✓ application
  - ✓ catch
  - ✓ **config** - šis ir galvenais konfigurācijas folderis
    - autoload.php
    - **config.php** - šeit var uzstādīt piem. bāzes tīmekļa adresi u.c. **uzstādījumus**
    - constants.php
    - **database.php** - šeit jākonfigurē datubāzes pieslēgšanās dati
    - doctypes.php
    - foreign\_chars.php
    - hooks.php
    - migration.php
    - mimes.php
    - profiler.php
    - **routes.php** - šeit var uzstādīt noklusēto kontrolieri u. c. **uzstādījumus**

- smileys.php
  - user\_agents.php
- ✓ **controllers** - šeit atrodas kontrolieri
- ✓ core
- ✓ errors
- ✓ **helpers** - šeit var definēt savus palīgus dažādu uzdevumu veikšanai
- ✓ hooks
- ✓ language
- ✓ **libraries** – šeit var definēt savas palīgbibliotēkas
- ✓ logs
- ✓ **models** - šeit atrodas modeļu php faili.
- ✓ third\_party
- ✓ **views** - šeit atrodas skatu php faili.
- ✓ system
  - ✓ core
  - ✓ database
  - ✓ fonts
  - ✓ **helpers** – šeit atrodas sistēmas palīgi dažādu uzdevumu veikšanai
  - ✓ language
  - ✓ **libraries** – šeit atrodas sistēmas palīgbibliotēkas
- ✓ user\_guide
  - **index.php**

### 7.3. att., CodeIgniter fiziskā struktūra

## 7.4. Datu bāzu pārvaldība un drošība

CodeIgniter atbalsta pieslēgšanos pie vairāk kā vienas datubāzes vienlaicīgi, ko var viegli realizēt. Ja programmētājs vēlas cīnīties ar SQL injekcijām, rakstot drošus SQL pieprasījumus, tad CodeIgniter ir parūpējies par to, izveidojot „Active Record” klasi, kas ļauj rakstīt drošus SQL pieprasījumus. Zemāk redzams „Active Record” bibliotēkas piemērs.

```

$this->db->select('Nosaukums, Atzime, Atzimes_apraksts');
$this->db->join('parbaudes_darbi','parbaudes_darbi.parbdarbi_ID = atzimes.parbdarbi_ID_FK');
$this->db->join('prieksmeti','prieksmeti.prieks_ID = parbaudes_darbi.prieks_ID_FK');
$query=$this->db->get_where('atzimes', array('Skolnieks' => $student_Isn, 'prieks_ID'=>$prieksmets_ID));
$item = $query->result();
$a=sizeof($items) ;
if($a>0)
{ //priekšmetā ielikto atzīmju izdruka, lietotjot ajax-metodi

```

### 7.4.1.att., Active Record

Sarežģītiem pieprasījumiem lieti noder arī „query binding”, kur tiek padots pieprasījums un arī dati, kuri jāliek „?” vietās, piemērs –

```
$sql = "SELECT * FROM some_table WHERE id = ? AND status = ? AND author = ?";  
$this->db->query($sql, array(3, 'live', 'Rick'));
```

#### 7.4.2.att., *Query binding*

Šādā veidā datubāze tiek pasargāta no SQL injekcijām, kas ļauj rakstīt sarežģītus un drošus SQL pieprasījumus. Šo pieprasījumu veidošana ir ērta un parocīga.

## 7.5.Ietvara iekšējā arhitektūra

CodeIgniter ir balstīts uz MVC strukturēšanas principa, kas nozīmē, ka CodeIgniter atbalsta atdalītu lietotnes biznesa loģiku no lietotnes prezentācijas. Var tikt veidoti skati, modeļi un kontrolieri. CodeIgniter ir samērā brīva pieeja MVC strukturēšanas principiem, tādā ziņā, ka nav uzspiesta obligāta modeļu lietošana. Ja šāda sadalīšana lietotnē nav vajadzīga, tad var likt datubāzes pārvaldības funkcijas (parasti to gan liek modeļos) arī kontrolieros. Skati parasti ir tīmekļa lapas, taču CodeIgniter pieļauj, ka skats var būt arī lapas fragments – galvene vai kājene. CodeIgniter pieļauj, ka skats var būt arī RSS lapa vai jebkāda cita tipa lapa. Ir iespējams veidot savas palīgbibliotēkas un iekļaut savus skriptus. Šāda brīva pieeja ļauj lietotājam strādāt pēc iespējas pierastākā, brīvākā veidā.

Balstoties uz MVC strukturēšanas principiem, no kontrolieriem tiek izsaukti skati un modeļi. Sintakse, lai izsauktu skatus un modeļus, ir viegli uztverama un saprotama. Kā to var realizēt CodeIgniter koda sintakses veidā, ir attēlots 7.5.1. attēlā.

```
function kontakti() // skolas kontaktu(informācijas) lapas sagatavošana  
{  
    $datas = $this->Register_model->skolas_kontakti($this->session->userdata('isn'),  
                                                    $this->session->userdata('skolas_ID') );  
    if($this->session->userdata('tips')==2)  
    {  
        $this->load->view("clienthash_password/student_kontakti", $datas );  
    }  
    if($this->session->userdata('tips')==3)  
    {  
        $this->load->view("clienthash_password/teacher_kontakti", $datas );  
    }  
}
```

#### 7.5.1.att., *Skatu un modeļu izsaukšana*

## 7.6. Ievadformas, to datu pārbaude un ģenerēšana

CodeIgniter ir parūpējies par ievadformu pārbaudes bibliotēku, ko var ērti ielādēt kontrolierī. Pēc tam var uzlikt vajadzīgos formu validācijas likumus un pārbaudīt, vai formas tiem atbilst. Validācijas likumus var uzstādīt ar ērtu sintaksi, kā attēlots zemāk esošajā attēlā.

```
<?php
class Form extends CI_Controller {

    function index()
    {
        $this->load->helper(array('form', 'url'));

        $this->load->library('form_validation');

        $this->form_validation->set_rules('username', 'Username', 'required');
        $this->form_validation->set_rules('password', 'Password', 'required');
        $this->form_validation->set_rules('passconf', 'Password Confirmation', 'required');
        $this->form_validation->set_rules('email', 'Email', 'required');

        if ($this->form_validation->run() == FALSE)
        {
            $this->load->view('myform');
        }
        else
        {
            $this->load->view('formsuccess');
        }
    }
}
?>
```

### 7.6. att., *Validācijas likumu veidošana*

Lai atvieglotu ievadformu ģenerēšanu, CodeIgniter ir radījis „Form” palīgu, kurā ir daudz noderīgas funkcijas, lai varētu arī ērti ģenerēt formas.

## 7.7. Speciālas iespējas

CodeIgniter ir pieejams plašs klāsts ar **palīgbibliotēkām**, kas ievērojami atvieglo un palīdz dažādos ierastos uzdevumos:

Calendar.php	- noder kalendāru veidošanā
Cart.php	- dots 'shopping cart' prototips
Driver.php	- ļauj pievienot klasēm papildus iespējas, lietojot draiveru objektus
Email.php	- ļauj sūtīt e-pastus, lietojot Mail, Sendmail vai SMTP
Encrypt.php	- ļauj veikt kriptogrāfijas uzdevumus
Form_validation.php	- ļauj pārbaudīt formas vienkāršā veidā
Ftp.php	- ļauj failus pārnest uz citu serveri
Image_lib.php	- dažādas noderīgas funkcijas bilžu apstrādes uzdevumiem
Javascript.php	- dažādas noderīgas javascript funkcijas
Log.php	- kļūdu saglabāšanas iespēja log failos
Migration.php	- datubāzes mainīšana migrācijas veidā
Pagination.php	- lapu numerācijas un pāriešanas pa lapām funkcijas
Parser.php	- sagataves parsēšana klase
Profiler.php	- profilēšanas klase, kas parāda SQL pieprasījums, \$_POST datus u. c. datus lapas apakšā, kas ir noderīgi lapas izstrādē
Session.php	- sesijas apstrādes funkcijas
Sha1.php	- Sha1 kriptogrāfijas funkcija
Table.php	- palīdz izveidot tabulas
Trackback.php	- ļauj saņemt un nosūtīt atsaučpaziņošanas datus
Typography.php	- palīdz formatēt tekstu
Unit_test.php	- vienkāršas testēšanas funkcijas
Upload.php	- failu augšupielādēšanas funkcijas
User_agent.php	- palīdz iegūt informāciju par interneta pārlūku, mobilo ierīci vai robotu, kas apmeklē lapu
Xmlrpcs.php	- XML-RPC pieprasījumu serveru puses apstrāde
Zip.php	- atļauj veidot zip arhīvus

Bez tam CodeIgniter ir arī daudzi **palīgi** (helpers) uzdevumu veikšanai, kuru funkcijas tiešām noder un atvieglo dzīvi. Tie lieti var noderēt skatos un kontrolieros. Eksistē šādi CodeIgniter palīgi: array\_helper, captha\_helper, cookie\_helper, date\_helper, directory\_helper, download\_helper, email\_helper, file\_helper, form\_helper, html\_helper, inflector\_helper, language\_helper, number\_helper, path\_helper, securiy\_helper, smiley\_helper, string\_helper, ext\_helper, typography\_helper, url\_helper, xml\_helper.

## **7.8.Trūkumi**

CodeIgniter ir ietvars, kuru, vienreiz uzinstalējot, ir samērā grūti atjaunināt, uzlikt jaunāku versiju, trūkst atbalsta to darīt. CodeIgniter modularitāte nav īpaši labā līmenī. Kods nav viegli sadalāms neatkarīgos moduļos, kas apgrūtina arī jaunu funkcionalitāšu pievienošanu un būtisku CodeIgniter pārbūvi, pielāgojot to savām vajadzībām. Kontrolieriem ir grūti uzrakstīt vienības testēšanas testus. CodeIgniter lieto vienu lielu globālo mainīgo (\$this->), kas nozīmē, ka grūti izmantot objekt-orientētās programmēšanas priekšrocības.

## 8. SYMFONY

PHP ietvars Symfony tika salīdzināts pēc salīdzināšanas kritērijiem.

### 8.1. Tehniskās prasības

Symfony 2.2 atbalsta PHP 5.3.3 un jaunākas versijas.

### 8.2. Uzstādīšana un jauna projekta izveide

Symfony uzstādīšana ir ātra un vienkārši izdarāma, ja izmanto bibliotēku „Composer” [8], kas ir domāts PHP atkarību sakārtošanai. „Composer” pats lejuplādē norādīto Symfony versiju un vajadzīgos paplašinājumus. Patīkami, ka Symfony uzinstalē arī parauga jaunus projektus, kuri iesācējiem tiešām palīdz izprast, kā procesi notiek šajā ietvarā.

### 8.3. Konfigurācijas pārvaldība

Symfony 2.2.0. konfigurācijas pārvaldība tiek organizē folderu struktūru un izkārtojumu, kas ir samērā intuitīvs – skatīt zemāk.

Symfony

- ✓ app - **šeit atrodas lietotnes konfigurēšanas faili**
  - ✓ config.yml
  - ✓ config\_dev.yml
  - ✓ config\_prod.yml
  - ✓ config\_test.yml
  - ✓ parameters.yml - **šeit var uzlikt datubāzes savienojuma datus**
  - ✓ routing.yml
  - ✓ routing\_dev.yml - **šeit var uzstādīt noklusēto kontrolieri u. c. uzstādījumus**
  - ✓ security.yml
- ✓ bin
- ✓ src – **šeit atrodas php faili, kurus raksta lietotājs**
- ✓
- ✓ vendor
- ✓ web - **šeit atrodas „root” folderis symfony ietvaram**

*8.3. att., Symfony fiziskā struktūra*

## 8.4. Datu bāzu pārvaldība un drošība

Var uzstādīt vairāku datubāzu atbalstu Symfony, kas nav grūti izdarāms [9], un, kas ir ļoti noderīgi lielu sistēmu izveidē. Symfony bibliotēka „Doctrine” vai projekts „Propel” atvieglo datubāzu pārvaldību ar ORM iespējām, kas pie viena izslēdz arī SQL injekciju iespējamību. Skatīt zemāk esošo attēlu, kurā attēlots bibliotēkā „Doctrine” realizētais ORM piemērs.

```
1 // src/Acme/StoreBundle/Controller/DefaultController.php
2
3 // ...
4 use Acme\StoreBundle\Entity\Product;
5 use Symfony\Component\HttpFoundation\Response;
6
7 public function createAction()
8 {
9     $product = new Product();
10    $product->setName('A Foo Bar');
11    $product->setPrice('19.99');
12    $product->setDescription('Lorem ipsum dolor');
13
14    $em = $this->getDoctrine()->getManager();
15    $em->persist($product);
16    $em->flush();
17
18    return new Response('Created product id '.$product->getId());
19 }
```

8.4.1.att., „Doctrine ORM” piemērs

Symfony ORM realizējums lieti noder un atvieglo darbu ar datubāzi, taču, ja ORM izmantošana neliekas labākais risinājums projektam, tad „Doctrine DBAL” bibliotēka piedāvā arī parastu SQL pieprasījumu rakstīšanu.

## 8.5. Ietvara iekšējā arhitektūra

Symfony arhitektūra ir balstīta uz MVC strukturēšanas principiem, tāpēc Symfony iekšējā arhitektūra sastāv no skatiem, kontrolieriem un modeļiem. Lietojot Symfony bibliotēku „Doctrine” vai projektu „Propel”, Symfony ar modeļiem saprot ORM kā datu reprezentētājus. Symfony ar skatiem saprot HTML, XML, CSV, LaTeX u. c. dokumentus, kurus var realizēt, lietojot php sagataves failus, vai arī var lietot

Symfony2 sagataves „Twig” iespējas. Skatīt zemāk attēlu ar Twig sagataves izmantošanu. Twig sintakse attēlā redzama 7., 10., 11.,12. rindiņā.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My Webpage</title>
5   </head>
6   <body>
7     <h1>{{ page_title }}</h1>
8
9     <ul id="navigation">
10      {% for item in navigation %}
11        <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
12      {% endfor %}
13    </ul>
14  </body>
15 </html>
```

#### 8.5.1.att., „Twig” piemērs

Symfony kontrolieris ir PHP funkcija, kas paņem informāciju no HTTP pieprasījuma un uzbūvē un atgriež HTTP atbildi, kas var būt HTML lapa, XML dokuments, attēls, 404 kļūda vai jebkas cits.

## 8.6.Ievadformas, to datu pārbaude un ģenerēšana

Symfony ievadformu pārbaude un ģenerēšana ir izveidota tiešām parocīga. Formu ģenerāciju likumus var norādīt atsevišķā folderī „Forms”, izveidojot formas klasi, kur var pievienot formas laukus, norādot, kādi pārbaudes likumi ir uz tiem attiecināmi (piemēram, e-pasta pārbaude u.taml.). Skatīt zemāk esošo attēlu.

```

1  <?php
2
3  namespace Acme\DemoBundle\Form;
4
5  use Symfony\Component\Form\AbstractType;
6  use Symfony\Component\Form\FormBuilderInterface;
7
8  class ContactType extends AbstractType
9  {
10     public function buildForm(FormBuilderInterface $builder, array $options)
11     {
12         $builder->add('email', 'email');
13         $builder->add('message', 'textarea');
14     }
15     public function getName()
16     {
17         return 'contact';
18     }
19 }

```

8.6.1.att., *Validācijas likumu veidošana*

## 8.7. Speciālas iespējas

Symfony ir pieejams plašs klāsts trešo pušu bundļu, kuras var importēt – tā iegūstot jaunu funkcionalitāti acu mirklī. Piemēram, šajā lapā [10] lietotāji ir iesūtījuši ap 2000 bundļu, kuras noder tādu funkcionalitāšu iegūšanai kā:

- CalendarBundle - kalendāra un kalendāra notikumu veidošanai
- KnpMenuBundle - objektorientētu meņu izveidei Symfony
- HWIOAuthBundle - atbalsta autentificēšanos, izmantojot twitter, facebook
- SwiftmailerBundle - atbalsta e-pastu nosūtīšanas iespējas,
- VichUploaderBundle - atbalsta failu augšupielādēšanas iespējas
- AvalancheImagineBundle - atbalsta dažādas manipulācijas ar bildēm
- NelmioJsLoggerBundle - atbalsta kļūdu rakstīšanu logos.
- ProfilerLiveBundle - atbalsta dažādus profilēšanas veidus
- KnpPaginatorBundle - lapu numerācijas un pāriešanas pa lapām funkcijas
- ManyMulesFontAwesomeBundle - palīdz formatēt tekstu ar dažādiem fontiem
- BCCronManagerBundle - atbalsta tabulu veidošanu
- PierreEncrypterBundle - atbalsta kriptogrāfijas iespējas
- JMSSerializerBundle - atbalsta datu serializēšanu XML, JSON, YAML.

- DataExporter - atbalsta datu eksportēšanu CSV, XML, JSON, Excel, HTML failos.

Vēl daudzas dažādas bundles praktiski visiem sistēmas uzdevumiem ir atrodamas Symfony direktoriju struktūrā.

## **8.8.Trūkumi**

Symfony neatbalsta vecākas PHP versijas kā PHP5. Symfony lieto daudz konfigurācijas failu, kas apgrūtinoši maziem projektiem. Symfony folderu struktūra liekas diezgan sarežģīta, salīdzinot ar citiem PHP ietvariem.

## 9. Yii

PHP ietvars Yii zemāk tiek salīdzināts pēc salīdzināšanas kritērijiem.

### 9.1. Tehniskās prasības

Yii ietvars atbalsta gan PHP 5.1.0 un jaunākas versijas.

### 9.2. Uzstādīšana un jauna projekta izveide

Kad Yii ietvara faili ir lejuplādēti un uzlikti uz servera, tad, lai izveidotu jaunu projektu, var izmantot konsoli, kas tiešām atvieglo un paātrina jauna projekta izveidošanu [11]. Vienīgi jāņem vērā, ka jauna projekta failiem jābūt virs Yii folderī esošā foldera ‘frameworks’, kas kalpo kā palīgbibliotēku un pamata bibliotēku folderis. Pozitīvi, ka šādā veidā var būt tikai viens ‘frameworks’ folderis un daudzi jaunu projektu folderi, kas visi izmanto ‘frameworks’ foldera bibliotēku.

### 9.3. Konfigurācijas pārvaldība

Yii 1.1.13 konfigurācijas pārvaldība tiek organizēta ar samērā viegli atšifrējamu folderu struktūru. Folderu struktūras attēlojumu var redzēt zemāk. Ir-izcelti tie folderi un faili, kuru lietošana un modificēšana būtu vajadzīga jauna projekta izveidei.

Yii

- ✓ framework
  - ✓ **yiic.bat** - šis failu ir jāpalaiž no konsoles, lai izveidotu jaunu projektu
  - ✓ **views** – dažādās valodās pieejami kļūdu skati
  - ✓ web
  - ✓ zii
  - ✓ test
  - ✓ utils
  - ✓ **validators** – noderīgas palīgbibliotēkas ievadlauku validācijai
  - ✓ vendors
  - ✓ **logging** – dažādi palīgfaili logu veidošanai
  - ✓ messages
  - ✓ gii
  - ✓ base
  - ✓ caching
  - ✓ cli
  - ✓ **collections** – dažādas vajadzīgas bibliotēkas kolekciju veidošanai
  - ✓ console
  - ✓ db
- ✓ Requirements

- **index.php** – palaižot šo failu, var redzēt, vai tiek izpildītas visas vajadzīgās prasības, lai lietotu Yii ietvaru.
- ✓ **new\_project** – jaunizveidotais projekts, kas veidots ar konsoles palīdzību
  - ✓ assets
  - ✓ **css** – šeit jāliek projekta css faili
  - ✓ **images** – šis folderis domāts projekta attēliem
  - ✓ protected – šeit atrodas sistēmas palīgi dažādu uzdevumu veikšanai
    - ✓ commands
    - ✓ components
    - ✓ config
    - ✓ **controllers** – šeit atrodas projekta kontrolieru php faili
    - ✓ data
    - ✓ extensions
    - ✓ messages
    - ✓ migrations
    - ✓ **models** – šeit atrodas projekta modeļu php faili
    - ✓ runtime
    - ✓ **tests** – šeit atrodas projekta testi
    - ✓ **views** – šeit atrodas projekta skatu php faili
  - ✓ themes
    - index.php
    - index-test.php

### 9.3. att., *Yii fiziskā struktūra*

## 9.4. Datu bāzu pārvaldība un drošība

Yii atbalsta vairāku datubāzu lietošanu vienlaicīgi, ko var panākt samērā ātri, sekojot instrukcijām [12]. Pozitīvi, ka Yii piedāvā vairākus veidus, kā veidot interakciju ar datubāzi. Var lietot Yii piedāvāto ORM izmantojošu tehniku - „Active Record”. Zemāk attēlā var redzēt „Active Record” piemēru.

```
$post=new Post;
$post->title='sample post';
$post->content='content for the sample post';
$post->create_time=time();
$post->save();
```

### 9.4.1.att., *Yii Active Record*

Ja „Active Record” tehnika neliekas piemērota, tad var lietot arī „Query Builder”, kas lieti noder, lai veidotu sarežģītus SQL vaicājumus. Zemāk attēlā var redzēt „Query Builder” piemēru.

```

$user = Yii::app()->db->createCommand()
->select('id, username, profile')
->from('tbl_user u')
->join('tbl_profile p', 'u.id=p.user_id')
->where('id=:id', array(':id'=>$id))
->queryRow();

```

#### 9.4.2.att., „Yii Query Builder” piemērs

Gan „Query Builder”, gan „Active Record” ir izveidoti, lai aizsargātu no SQL injekcijām. Patīkami, ka Yii piedāvā plašu izvēli, kā modificēt datubāzi.

## 9.5.Ietvara iekšējā arhitektūra

Tā kā Yii ir balstīts uz MVC strukturēšanas principa, tad tas automātiski nozīmē, ka Yii iekšējā arhitektūra sastāv gan no skatiem, gan no modeļiem un kontrolieriem. Yii ietvara kontekstā modelis nozīmē vienu datu objektu. Tas var reprezentēt gan rindu datubāzes tabulā, gan HTML formu ar lietotāja ievadlaukiem. Katrs datu objekta lauks tiek reprezentēts ar kādu modeļa atribūtu. Atribūtam ir nosaukums, un tas var tikt pārbaudīts ar definētiem likumiem. Formu gadījumā tā ir ievadlauku pārbaude. Ar skatiem Yii saprot lietotāja interfeisa elementus, kas parasti ir PHP faili. Yii ir iespēja izmantot Layout elementus, kas atvieglo galveņu un kājeņu izmantošanu, kā arī ir iespēja izmantot patstāvīgus *Widget* elementus – tādus sarežģītus objektus kā kalendāri u. tml. Kontrolieru uzdevums ir paveikt pieprasīto darbību, kas parasti ir vajadzīgā modeļa iegūšana un vajadzīgā skata atgriešana, skatīt zemāk attēlu, kurā attēlota kontrolierī esoša funkcija.

```

public function actionApprove()
{
    if(Yii::app()->request->isPostRequest)
    {
        $comment=$this->loadModel();
        $comment->approve();
        $this->redirect(array('index'));
    }
    else
        throw new CHttpException(400,'Invalid request');
}

```

#### 9.5.1.att., Yii kontroliera piemērs

## 9.6. Ievadformas, to datu pārbaude un ģenerēšana

Yii ievadformu ģenerēšana un datu pārbaudes likumu izveidošana ir samērā viegli veicama. Formas ar ievadlaukiem un to ģenerēšanas likumus var definēt kā modeļus, kas ļauj ievadformas ērti izmantot vairākās vietās, definējot tās tikai vienā vietā [13]. Yii piedāvā arī iebūvētu bibliotēku – „CactiveForm”, kas atrodas zem foldera „Framework” un var lieti noderēt, ģenerējot ievadformas.

## 9.7. Speciālas iespējas

Yii ir pieejamas plašs klāsts ar **palīgibibliotēkām**, kas ievērojami atvieglo un palīdz dažādos ierastos uzdevumos.

Ir pieejamas šādas direktorijas, kuros atrodamās bibliotēkas palīdz dažādos uzdevumos:

Validators	- noder ievadlauku validācijā
Logging	- noder efektīvu logu veidošanā
Gii	- koda ģenerators, kas ļauj ģenerēt modeļus utt.
collections	- atvieglo dažādu objektu kolekciju veidošanu
console	- ļauj lietot yii konsoli, ar kuru iespējams ģenerēt jaunus projektus un arī kodu un veikt vēl citus uzdevumus
Utils	- šajā direktorijā atrodas bibliotēkas, kuras palīdz tādos uzdevumos kā failu augšupielādes, lejupielādes, datuma formatēšana, nezināma lieluma masīva izprintēšana u. c.
i18n	- internacionalization - palīdz ar sagatavotām bibliotēkām kļūdas paziņojumu, ciparu, datu iegūšanai no faila u. c. gadījumos daudzās valodās.

## 9.8. Trūkumi

Yii ietvara dokumentācija brīžiem šķiet neērta un diezgan virspusēja. Pieejamās Yii bibliotēkas un papildiespējas atrodas dažādos folderos, dažādos līmeņos, kas palielina iespēju, ka tās var tikt neatrastas un pilnībā neizmantotas.

## 10. PROJEKTI UN TIEM ATBILSTOŠU IETVARU IZVĒLE

Kad ir zināmi projekta uzdevumi un ir izvēlēta PHP programmēšanas valoda, kurā realizēt projektu, tad ir jāizvēlas kādu ietvaru un vai vispār kādu ietvaru lietot.

### 10.1. Teorētisks projektu apskats

Šajā projektu apskatā tiks ieteikti 4 šobrīd paši populārākie jau iepriekšējās nodaļās apskatītie ietvari - CodeIgniter, Yii, CakePHP, Symfony. Ņemot vērā projektu prasības, izvēloties piemērotu ietvaru vai vispār izvēloties ietvaru nelietot, projektējamajai sistēmai ir jāņem vērā šādi kritēriji:

#### 10.1.1. Sistēmas lielums

Projektējamās sistēmas lielums ir svarīgs faktors, lai izvēlētos piemērotāko ietvaru.

- **Liela sistēma**

Lielu sistēmu izveides gadījumā ieteicams izvēlēties - Yii, CodeIgniter, Symfony[14]. CakePHP nav ieteicams izmantot lieliem projektiem, jo CakePHP lielu sistēmu gadījumā darbojas salīdzinoši lēni lielās automatizācijas, kas ir šajā ietvarā, dēļ.

- **Vidēji liela sistēma**

Vidēji lielu sistēmu gadījumā ieteicams izvēlēties Yii, CodeIgniter vai CakePHP. Šajā sarakstā Symfony nav iekļauts, jo šim ietvaram ir ļoti daudz konfigurējamu failu, kas ir izdevīgi tikai lielu sistēmu gadījumā.

- **Maza sistēma**

Mazu sistēmu gadījumā ieteicams lietot Yii, CodeIgniter, CakePHP ietvarus, jo šie ietvari ļauj mazām sistēmām būt ātrām un vienkārši pārvaldāmām, kas ir PHP ietvaru galvenais uzdevums.

### 10.1.2. Sistēmas nākotnes perspektīvas

Sistēmas nākotnes perspektīvas apraksta to, vai

- **sistēma var tikt paplašināta/radikāli mainīta vajadzības gadījumā**  
Šādā gadījumā ieteicams lietot Symfony, Yii vai CakePHP, jo šie ietvari atbalsta modularitāti, kas nozīmē, ka ietvars sastāv no moduļiem. Vajadzības gadījumā var pievienot papildus moduļus, neizjaucot jau pašreizējo moduļu darbību. CodeIgniter ietvars neatbalsta modularitāti.
- **sistēmai nevajag būt paplašinātai**  
Šādā gadījumā ieteicams lietot Symfony, Yii, CodeIgniter vai CakePHP.

### 10.1.3. Sistēmas kvalitāte

Sistēmas kvalitāte nozīmē to, cik viegli ir uzturēt sistēmu, kā arī to, cik viegli jauns programmētājs var apgūt jau uzbūvēto sistēmu. Tā kā visi četri apskatītie ietvari atbalsta MVC strukturēšanas principu, tad var izvēlēties jebkuru no šiem četriem ietvariem – Symfony, Yii, CodeIgniter vai CakePHP.

### 10.1.4. Sistēmas ātrdarbība

Sistēmas ātrdarbība ietvariem ir svarīgs faktors, jo PHP ietvari sastāv no daudzām bibliotēkām, un, lai izsauktu vienu pieprasījumu, ietvars ielādē un izpilda vairākus PHP failus. Pēc pētījumiem [15] [16] [17], kuros apskatītas dažādas ietvaru versijas, atklājas, ka daudzos no tiem visātrākais ir Yii ietvars. Ātruma ziņā otrais labākais ir CodeIgniter. Pēc tam – CakePHP un tad Symfony. Daudzie konfigurācijas faili varētu izskaidrot, kāpēc Symfony ietvars ir salīdzinoši lēnāks nekā citi ietvari.

## 10.2. Praktisks projektu apskats

Šajā nodaļā aprakstīta pieredze ar projektiem, kuros jāizvēlas, kādu PHP ietvaru lietot. Šie projekti ir minēti zemāk.

### 10.2.1. Tīmeklī bāzēta skolvadības sistēma

Sistēmas funkcionalitāte lielos vilcienos ir šāda (aprakstīta pa lomām)

Skolēni

- ✓ apskata savas atzīmes
- ✓ iesūta mājasdarbus

Skolotāji

- ✓ pārvalda pārbaudes darbu savos priekšmetos
- ✓ pārvalda skolēnu atzīmes pārbaudes darbos
- ✓ pārvalda skolēnu iesūtītos mājasdarbus
- ✓ pārvalda audzināmās klases un klašu paziņojumus

Administratori

- ✓ pārvalda skolotājus un skolotāju priekšmetus
- ✓ pārvalda klases un skolēnus tajās
- ✓ pārvalda skolas paziņojumus

### 10.2.2. Kādu PHP ietvaru izvēlēties skolvadības sistēmai?

Ņemot vērā „Skolvadības” sistēmas PPS funkcionālās un nefunkcionālās prasības, ir izvēlēts tīmekļa ietvars CodeIgniter, jo CodeIgniter ir viens no ātrākajiem tīmekļa ietvariem. Tam ir ļoti laba un plaša dokumentācija un tā uzbūve ir samērā vienkārša, taču reizē arī labi strukturēta pēc labākajiem MVC principiem. CodeIgniter labā dokumentācija ļauj iesācējam programmētājam diezgan ātri apgūt MVC principus un visu kontrolieru, modeļu un skatu kopsakarības koda ziņā, savukārt CodeIgniter URL veidošana ļauj veidot uzreiz klientam pieņemamas un skaistas tīmekļa adreses, pat neveicot URL pārrakstīšanu.

Zemāk var redzēt ekrānšāviņus ar uzbūvēto sistēmu, lietojot CodeIgniter ietvaru.

Jūs esat pieslēdzies kā Juris Liepiņš!  
Izlogoties āra

Galvenā lapa Skolotāji Klases, skolēni Profils

Izveidot jaunu skolotāju

Nr.	Skolotājs	Māca priekšmetus	Dzēst skolotāju
1.	Zane Cilinska	Spied šeit	X
2.	Jānis Jubalts	Spied šeit	X
3.	Mārtiņš Zemzars	Spied šeit	X

Priekšmeta nosaukums	Klase	Dzēst priekšmetu
Angļu valoda	7.d	X
Latīņu valoda	9.a	X

Pievieno priekšmetu , ko māca šis skolotājs! Aizpildi laukus un spied - "Pievienot!"

Priekšmets :  Klase :

Aizvērt

Ievadi jauna skolotāja profila datus

Lietotājvārds \*

Vēlamā parole \*

Vēlamā parole vēlreiz \*

Vārds \*

Uzvārds \*

Ievadiet skolotāja e-pasta adresi, uz kuru tiks nosūtīti profila lietotājvārds un parole!  
Vairākas e-pasta adreses, atdaliel ar komatu!

E-pasts \*

10.2.2.att., Skolvadības sistēmas attēls - 1

Attēlos var redzēt, ka CodeIgniter ietvars ir tiešām piemērots skolvadības sistēmai.

Galvenā lapa Priekšmeti Profils Skolas kontakti

Nr.	Priekšmets	Klase
1.	Ķīmija	12.a

Nr.	Pārbaudes darba nosaukums	Dzēst pārbaudes darbu
1	1.md	X
2	2.md	X
3	3.md	X

Pārbaudes darba apraksts: Izrēķināt ķīmijas uzdevumus, lietojot MS Excel.  
Iesniegt '.xls' failu!  
Šim pārbaudes darbam ir pievienota mājasdarba augšuplādes iespēja!  
Mājasdarba augšuplādēšanas termiņš : 2011-12-05 12:00:00

Skolēna vārds uzvārds	Atzīme	Atzīmes apraksts	Iesūtītais mājasdarbs
Pēteris Ūsiņš	9	Malacis! Teicami.	Skatīt

Iesūtīts: 2011-06-01 13:59:09  
Lejuplādēt Book1-1.xls

Skolēna vārds	Atzīme	Iesūtītais mājasdarbs
Anna Vālodze	8	Skatīt
Laima Putniņa	8	Skatīt
Ivars Jankovskis	7	Skatīt

Pievienot pārbaudes darbu šim priekšmetam!

10.2.3.att., Skolvadības sistēmas attēls - 2

### 10.2.3. Ceļojumu firmas sistēmas sagataves izveidošanas apraksts

Sistēmas funkcionalitāte lielos vilcienos ir aprakstīta zemāk, pa lomām.

Viesi

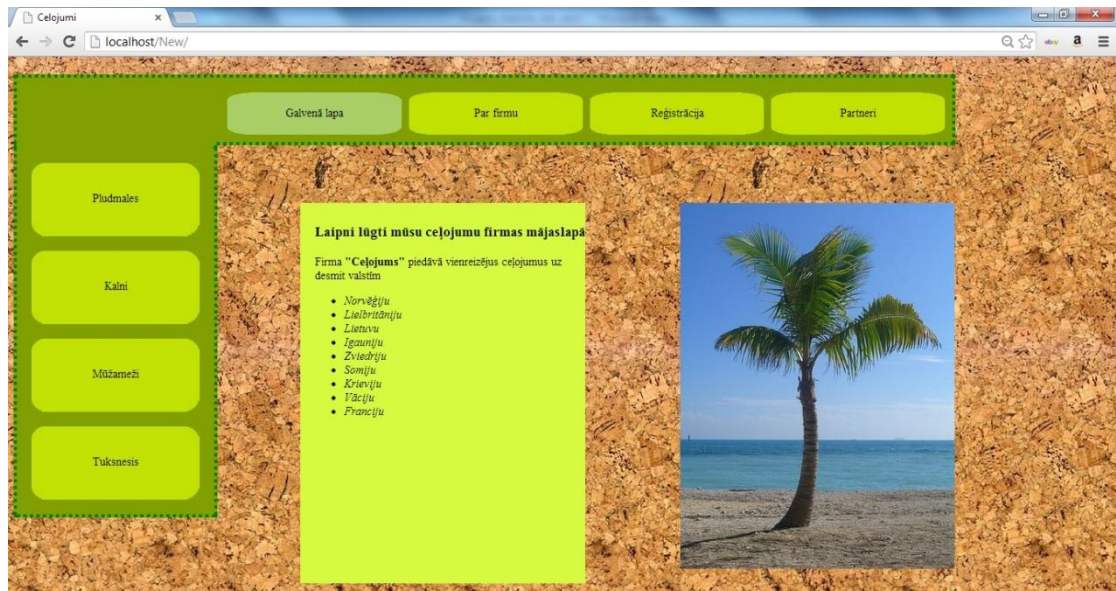
- ✓ apskata ceļojumu firmas informāciju
- ✓ var reģistrēt savu e-pastu, lai saņemtu informāciju par jaunākajiem ceļojumu firmas piedāvājumiem

Administratori

- ✓ pārvalda ceļojumu firmas piedāvājumus
- ✓ pārvalda ceļojumu firmas visas lapas

### 10.2.4. Kādu PHP ietvaru izvēlēties ceļojumu sistēmas izstrādei?

Ņemot vērā to, ka ir jāsagatavo ceļojumu firmas sistēmas sagatave, kas nozīmē, ka tā vēlāk tiktu uzlabota un pārveidota, tad izvēlēts ietvars, kura struktūra sastāv no moduļiem un kurš ir viens no visātrākajiem ietvariem – Yii.



10.2.4.att., *Ceļojumu firmas sistēmas sagataves attēls*

## REZULTĀTI

Bakalaura darba „PHP IETVARU SALĪDZINĀJUMS UN IZVĒLE

ATKARĪBĀ NO PROJEKTU MĒRĶIEM” ietvaros tika pētīti, salīdzināti pašlaik populārākie PHP tīmekļa ietvari – CakePHP, CodeIgniter, Symfony un Yii. Tika rūpīgi salīdzināti pētāmie ietvari pēc vienotiem kritērijiem - ietvara uzstādīšanas un jauna projekta izveides vieglums, konfigurācijas pārvaldība, ievadformu datu pārbaude un ģenerēšana, datubāzes pārvaldību, pieejamās ietvaru palīgbibliotēkas u. c. Gala rezultātā pēc ietvaru salīdzinājuma rezultātiem apskatīti dažādi projekti ar dažādām īpašībām un izsvērts, kādi ietvari ir vislabāk piemēroti dažādu projektu mērķiem.

## SECINĀJUMI

Izstrādājot darbu, secināts, ka visi pašlaik populārākie apskatītie ietvari ir labi katrs savā ziņā un var noderēt dažādu projektu mērķu sasniegšanā. Katram ietvaram ir savi plusi un mīnusi.

CodeIgniter ir ātrs, viegli apgūstams PHP ietvars. Tas ir lieliski piemērots programmētājiem pamatīgās un izsmeļošās dokumentācijas dēļ un noder mazām, vidējām un lielām sistēmām, kas jāizstrādā projekta vajadzībām. Vienīgi tas nesastāv no moduļiem, tāpēc nav piemērots projektiem, kas paredz lielu mainību izstrādājamajās sistēmās.

CakePHP ir pamatīgi izstrādāts, ar striktām konvencijām, kas automatizē un paātrina daudzu uzdevumu veikšanu. Tas var noderēt mazu un vidēja lieluma sistēmu izstrādē. CakePHP sastāv no moduļiem, tāpēc ir piemērots projektiem, kas paredz lielu mainību izstrādājamajās sistēmās.

Yii ietvars ir vienkāršs, drošs, samērā viegli apgūstams. Yii ir piemērots visa veida lieluma ietvaram. Yii joprojām nemainīgi augošā popularitāte garantē, ka tas tiks uzturēts vēl ilgu laiku. Yii arī ir modulārs ietvars.

Symfony ir apjomīgs, ar daudzām bibliotēkām apveltīts ietvars, kurš ir vairāk piemērots projektiem, kuros izstrādā liela izmēra sistēmas, jo daudzo konfigurācijas failu dēļ ātrdarbība ir lēnāka, salīdzinot ar citiem ietvaram, taču Symfony modularitāte nodrošina, ka vienmēr varēs kodu papildināt ar jauniem moduļiem, savukārt Symfony kvalitatīvā dokumentācija atvieglo ikviena pūles, lai apgūtu šo ietvaru.

Bakalaura darbam izvirzītais mērķis sasniegts. Bakalaura darba izstrāde par PHP ietvaram devusi daudz jaunu zināšanu, ko var aktīvi izmantot programmēšanas praksē.

## IZMANTOTĀS LITERATŪRAS SARAKSTS

1. „History of PHP” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://www.php.net/manual/en/history.php.php>
2. „Google Trands” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://www.google.com/trends/>
3. „Top 10 Hot PHP Frameworks” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://www.phpframeworks.com/top-10-php-frameworks/>
4. „Popular PHP Frameworks” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://blog.dayspring-tech.com/blog/2008/01/16/popular-php-frameworks-symfony-cakephp-codeigniter/>
5. „Model–view–controller” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
6. „CakePHP Cookbook” [tiešsaiste] - [skatīts 03.06.2013.]  
Pieejams: <http://book.cakephp.org/2.0/en/index.html>
7. „CodeIgniter” user guide [tiešsaiste] - [skatīts 03.06.2013.]  
Pieejams: [http://codeigniter.com/user\\_guide/](http://codeigniter.com/user_guide/)
8. „Installing and Configuring Symfony” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://symfony.com/doc/master/book/installation.html>
9. „Multiple Entity Managers” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams:  
[http://symfony.com/doc/2.0/cookbook/doctrine/multiple\\_entity\\_managers.html](http://symfony.com/doc/2.0/cookbook/doctrine/multiple_entity_managers.html)
10. „Knp Bundles” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://knpbundles.com/>
11. „Creating Your First Yii Application” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://www.yiiframework.com/doc/guide/1.1/en/quickstart.first-app#connecting-to-database>
12. „Multiple-database support in Yii” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://www.yiiframework.com/wiki/123/multiple-database-support-in-yii/>

13. „Creating Forms with the Yii Framework” [tiešsaiste] – [skatīts 03.06.2013]  
Pieejams: <http://www.larryullman.com/2011/01/20/creating-forms-with-the-yii-framework/>
14. „Top 5 PHP Frameworks Infographic” [tiešsaiste] – [skatīts 03.06.2013.]  
Pieejams: <http://php.dzone.com/articles/top-5-php-frameworks>
15. „Symfony faster than Yii” [tiešsaiste] – [skatīts 03.06.2013]  
Pieejams: [http://www.yiiframework.com/forum/index.php/topic/10198-symfony-2-faster-then-yii/page\\_st\\_20](http://www.yiiframework.com/forum/index.php/topic/10198-symfony-2-faster-then-yii/page_st_20)
16. **Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre** „*Programming PHP*”, 2008.
17. **Julie C. Meloni** „*PHP Essentials, 2nd Edition*”, 2010.

## PIELIKUMI

Šajā nodaļā ir iekļauti Tīmeklī Bāzētas Skolvadības Sistēmas programmatūras koda fragmenti, kas realizēti PHP tīmekļa ietvarā CodeIgniter, kuri ir no dažādiem moduļiem un atšķiras ar savu funkcionalitāti.

### 1. Student .php(kontrolieris)

```
<?php
/*****
Autors: Krišjānis Kokorīts
Nosaukums: student.php
Modulis: Student
Funkcijas:
    mājasdarbu apstiprināšana vērtēšanai, priekšmetu parādīšana , iesniegto mājasdarbu dzēšanu ,
    mājasdarbu iesniegšanas vai lejupielādes skata parādīšana, mājasdarba iesniegšana , atzīmju parādīšana ,
    priekšmetu skolotāju kontaktu parādīšana , priekšmeta mājasdarbu parādīšana.
*****/

class Student extends Controller {

    function Student()
    {
        parent::Controller();
    }

    function __construct()
    {
        parent::Controller();
        $this->load->library('session');
        $this->load->helper('form');
```

```

    $this->load->helper('url');
    $this->load->model("Student_model");
    set_time_limit( 300 );
}

function index()
{
    redirect("");
}

function apstiprinat_failu_vertesanai($md_faili_ID) // mājasdarbu apstiprināšana vērtēšanai
{
    $this->Student_model->apstiprinat_failu_vertesanai($md_faili_ID,$this->session->userdata('isn') ) ;
}

function delete_file($md_faili_ID) // iesniegto mājasdarbu dzēšanu
{
    $this->Student_model->delete_file($md_faili_ID,$this->session->userdata('isn') ) ;
}

function atzimes() // atzīmju parādīšana
{
    $datas =$this->Student_model->atzimes($this->session->userdata('isn') ) ;
    $this->load->view("clienthash_password/student_atzimes", $datas) ;
}

function upl_or_downl($parb_ID) // mājasdarbu iesniegšanas vai lejupielādes skata parādīšana
{
    $datas =$this->Student_model->upl_or_downl($this->session->userdata('isn'),$parb_ID) ;
}

function do_upload_md($parb_ID) // mājasdarba iesniegšana - ajax funkcija
{
    $lietotajs=$this->session->userdata('isn');
    $file=pathinfo($_FILES['userfile']['name']);
    $extension=$file['extension'];

```

```

$filename=$parb_ID.$lietotajs.'.'.$extension ;
$pazinojums=array('pazinojums');
$config['upload_path'] = './majasdarbi/';
$config['allowed_types'] = 'pdf|zip|doc|docx|jpg|gif|ppsx|pps|xls|xlsx|exe|rar|txt'; //augšupielādei atļautie
$config['max_size'] = '9999999999'; // m.d. tipi
$config['max_width'] = '10240';
$config['max_height'] = '76800';
$config['file_name'] = $filename;
$config['source']['max_size'] = '9999999999';
$config['overwrite'] =true ;
$this->load->library('upload', $config);
if ( ! $this->upload->do_upload() )
{
    $pazinojums['pazinojums'] = $this->upload->display_errors();
}
else
{
    $klases_ID=$this->session->userdata('klases_ID') ;
    $md_saved=$this->Student_model->do_upload_md($lietotajs,$klases_ID,$parb_ID, $_FILES['userfile']['name'] );
    if ($md_saved)
    {
        $pazinojums["pazinojums"]="'Augšupielādēts'" ;
    }
    else
    {
        $pazinojums["pazinojums"]="'Fails ir veiksmīgi augšuplādēts, taču nav saglabāts datubāzē!'" ;
    }
}
echo '<html><head> <script language="javascript" type="text/javascript">
    window.top.window.PutMessage('.$pazinojums["pazinojums"].','.$parb_ID. ');
</script>
</head>
<body>
</body>
</html>' ; //Javascript funkcijas izaukšana, paziņojumu izlikšanai iframe .

```

```

    }

    function load_works($prieks_ID) // mājasdarbu parādīšana priekšmetā
    {
        $this->Student_model->load_works($prieks_ID,$this->session->userdata('isn')) ;
    }

    function load_atzimes_ajax($prieksmets_ID) // atzīmju parādīšana
    {
        $this->Student_model->load_atzimes_ajax2($prieksmets_ID) ;
    }

    function load_teacher_info($prieksmets_ID,$teacher_ID) // priekšmetu skolotāju kontaktu parādīšana
    {
        $this->Student_model->load_teacher_info($prieksmets_ID,$teacher_ID) ;
    }
}

```

## 2. Student\_model.php(modelis)

```
<?php
```

```

/*****

```

```
Autors: Krišjānis Kokorīts
```

```
Nosaukums: student_model.php
```

```
Modulis: Student
```

```
Funkcijas:
```

```
    mājasdarbu apstiprināšana vērtēšanai, iesniegto mājasdarbu dzēšana ,
```

```
    priekšmetu apskatīšanās iespēja, mājasdarba augšupielāde ,
```

```
    mājasdarbu iesniegšanas vai lejupielādes skata parādīšana, atzīmju parādīšana,
```

```
    priekšmetu skolotāju kontaktu parādīšana.
```

```

*****/

```

```

class Student_model extends Model {

```

```

public function __construct()
{
    parent::__construct();
    $db['default']['char_set'] = "utf8_latvian_ci";
    $this->load->database();
}

public function atzimes($isn) //priekšmetu parādīšana
{
    $datas= array("prieksmets") ;
    $klases_ID=$this->session->userdata('klases_ID') ;
    $this->db->select('prieksmeta_nosaukums,Skolotajs, Name,Surname,prieks_ID');
    $this->db->join('prieksmeti','klases.klases_ID = prieksmeti.klase_parb');
    $this->db->join('registered_users','prieksmeti.Skolotajs = registered_users.isn');
    $query=$this->db->get_where('klases', array('klases_ID' => $klases_ID));
    $items = $query->result();
    $a=sizeof($items) ;
    if($a>0)
    {
        //Priekšmetu salikšana masīvā, lai pēc tam ielādētu skatā
        $i=$g=0 ;
        do
        {
            $g=$g+5 ;
            $datas["prieksmets"] []=$items[$i]->prieksmeta_nosaukums ;
            $datas["prieksmets"] []=$items[$i]->Name ;
            $datas["prieksmets"] []=$items[$i]->Surname ;
            $datas["prieksmets"] []=$items[$i]->Skolotajs ;
            $datas["prieksmets"] []=$items[$i]->prieks_ID ;
            $datas["prieksmets"] ['total']=$g ;
            $i++ ;
        }while($i<$a) ;
    }
    else
    {
        $datas["pazinojums2"]="Jums nav prieksmeti pieskirti / iespējams kluda ar pieslegsanos datuibaazei" ;
    }
}

```

```

    }
    return $datas ;
}

public function load_atzimes_ajax2($prieksmets_ID) // atzīmju parādīšana
{
    $student_Isn=$this->session->userdata('isn');
    $klases_ID=$this->session->userdata('klases_ID') ;
    $this->db->select('Nosaukums, Atzime, Atzimes_apraksts');
    $this->db->join('parbaudes_darbi','parbaudes_darbi.parbdarbi_ID = atzimes.parbdarbi_ID_FK');
    $this->db->join('prieksmeti','prieksmeti.prieks_ID = parbaudes_darbi.prieks_ID_FK');
    $query=$this->db->get_where('atzimes', array('Skolnieks' => $student_Isn, 'prieks_ID'=>$prieksmets_ID ));
    $items = $query->result();
    $a=sizeof($items) ;
    if($a>0)
    { //priekšmetā ielikto atzīmju izdruka(vajadzīgs ajax-metodei).
        $i=$g=0 ;
        echo '<table border="1" width="560" ><tr>
        <th >Parb. darba nosaukums</th>
        <th>Atzime</th>
        <th>Skolotāja komentārs</th>
        </tr> ' ;
        do
        {
            $g=$g+3 ;
            echo '<tr>
            <td >'. $items[$i]->Nosaukums . ' </td>
            <td> ' . $items[$i]->Atzime.' </td>
            <td> ' . $items[$i]->Atzimes_apraksts . ' </td>
            </tr>';
            $i++ ;
        }while($i<$a) ;
        echo '</table>';
    }else
    {
        echo "Atzīmes šajā priekšmetā nav ieliktas!" ;
    }
}

```

```

    }
}

public function load_works($prieks_ID,$lietotajs) // priekšmeta mājasdarbu parādīšana
{
    $sql = "select `parbdarbi_ID`, `Nosaukums`,`Lidz_kuram`,`augsupladet`,`parbdarba_apraksts` from prieksmeti
    JOIN parbaudes_darbi
    ON parbaudes_darbi.`prieks_ID_FK` = prieksmeti.`prieks_ID`
    where NOT `parbdarba_apraksts` = ' '
    and prieks_ID=? ;" ;
    $query=$this->db->query($sql, array($prieks_ID));
    $items = $query->result();
    $a=sizeof($items) ;
    if($a>0)
    {
//Visu mājasdarbu priekšmetā izdrukāšana tabulas veidā (vajadzīgs ajax funkcijai)
        $i=0 ;
        echo '<table width="560" ><tr>
        <th >Mājasdarba nosaukums</th>
        <th>Termiņš</th>
        </tr> ' ;
        do
        {
            if ($items[$i]->augsupladet!=1) //ja pārbaudes darbam nav md. augšupielādes iespēja uzlikta
            {
                echo '<tr>
                <td Onclick="upl_or_downl('.$items[$i]->parbdarbi_ID .')" >
                <a class="TableLinks">'. $items[$i]-> Nosaukums. '</a> </td>
                <td> ' .$items[$i]->Lidz_kuram.' </td>
                </tr>';
                echo '<tr class="hide_row" id="upload_'. $items[$i]->parbdarbi_ID.'" >
                <td colspan="2">Mājasdarba apraksts: <i>'. $items[$i]->parbdarba_apraksts.'</i>
                <div id="upload_div'. $items[$i]->parbdarbi_ID.'"> </div></td>
                </tr> ' ;
            }
        }
    }
}

```

```

else //ja pārbaudes darbam ir md. augšupielādes iespēja uzlikta, tad parāda faila iesniegšanas,
{
    //skatu
    echo '<tr>
    <td Onclick="upl_or_downl('.$items[$i]->parbdarbi_ID .)' " >
    <a class="TableLinks">'. $items[$i]-> Nosaukums. '</a>
    <a class="error"> Iesūtīt!</a>
    </td>
    <td> ' .$items[$i]->Lidz_kuram.' </td>
    </tr>';
    echo '<tr class="hide_row" id="upload_'. $items[$i]->parbdarbi_ID.'" >
    <td colspan="2">Mājasdarba apraksts: <i>'. $items[$i]->parbdarba_apraksts.'</i>
    <div id="upload_div'. $items[$i]->parbdarbi_ID.'"> </div></td>
    </tr> ' ;
}
$i++;
}while($i<$a) ;
echo '</table>';
}else
{
    echo "Mājasdarbi nav atrasti!<br/>" ;
}
}

public function do_upload_md($lietotajs,$klases_ID,$parb_ID, $filename) // mājasdarba augšupielāde
{
    $this->db->query("call insert_md_darbi($lietotajs,$klases_ID,$parb_ID,'$filename' )");
    $affected=$this->db->affected_rows();
    if ($affected==1)
    {
        return true;
    }else
    {
        return false ;
    }
}

public function apstiprinat_failu_vertesanai($md_faili_ID, $lietotajs ) // mājasdarbu apstiprināšana vērtēšanai

```

```

{
//ierakstišana datubāzē, ka mājasdarbs ir apstiprināts vērtēšanai
    $sql = "UPDATE
        md_faili
        JOIN
        registered_users
        ON
        registered_users.isn = md_faili.Skolnieks_isn
        SET
        md_faili.Apstiprinats =1
        WHERE
        registered_users.isn= ?
        and
        md_faili.md_faili_ID=?
        " ;
    $query = $this->db->query($sql, array($lietotajs, $md_faili_ID));
    $affected= $this->db->affected_rows();
    if ($affected==1)
    {
        echo "Fails iesniegts vērtēšanai" ;
    }else
        echo "Fails netika iesniegts vērtēšanai" ;
    }

    public function upl_or_downl($lietotajs,$parb_ID ) // mājasdarbu iesniegšanas vai lejupielādes skata parādīšana
    {
        $query= $this->db->query("call select_faili($parb_ID,$lietotajs)");
        $items = $query->result();
        $a=sizeof($items) ;
        if ($a>0)
        {
            if ($items[0]->augsupladet)
            {
                $b= date_default_timezone_set("Europe/Helsinki");
                $Currentdata= date('Y-m-d').' '.date("H:i:s") ; //pašreizējais laiks
                $Lidz_kuram=$items[0]->Lidz_kuram ;
            }
        }
    }
}

```

```

$access=true ;
$pazinojums=" " ;
if ($Lidz_kuram!='0000-00-00 00:00:00') //Ja laiks ir uzstādīts,
{
    tad pārbauda, vai iesniegšanas termiņš nav pagājis
    $access=$this->greaterDate($Lidz_kuram, $Currentdata) ;
}
if (!$access) //ja iesniegšanas termiņš nav pagājis, nevar iesniegt m.d.
{
    $pazinojums= "Iesniegšanas termiņš ir pagājis!</p>" ;
}
if(isset($items[0]->md_faili_ID))
{
    echo 'Iesūtīts: '.$items[0]->Iesutisanas_laiks.'<br/>
    <a href="'.base_url().'index.php/portal/homework_download/'.
    $items[0]->Faila_nosaukums.'/'.
    $items[0]->parbdarbi_ID_FK.'/0/' >
    Lejuplādēt <b>'.$items[0]->Faila_nosaukums.'</b>
    </a>' ;
    if (!$items[0]->Apstiprinats))
    {
        $pazinojums="Apstipriniet failu vērtēšanai!" ;
        echo '<a href="javascript:apstiprinat_vertesanai('.$items[0]->md_faili_ID.')" >
        <br/>Apstiprināt failu vērtēšanai.</a>' ;
        echo '<a href="javascript:delete_file('.$items[0]->md_faili_ID.','.$parb_ID.')" >
        <br/>Dzēst failu.</a>' ;
    }else
    $pazinojums= "Fails ir iesniegts un apstiprināts vērtēšanai!";
    echo '<p class="error" id="failu_paz_'.$items[0]->md_faili_ID.'">'.$pazinojums .' </p>';
}else
{
    if ($access) //ja iesniegšanas termiņš nav pagājis, var iesniegt majasdarbu, parādas augšupielādes
    {
        //forma
        echo '<form action="'.base_url().'index.php/student/do_upload_md/'.$parb_ID.'"
        method="post" enctype="multipart/form-data" target="for_ajax" >
        <input type="file" name="userfile" size="20" />
        <br /><br />
    }
}

```

```

        <input type="submit" value="Augšupielādēt" />
    </form> ' ;
    }
    echo '<p id="forma". $parb_ID.'" class="error" >'. $pazinojums .' </p>';
    }
    }
}

public function delete_file($md_faili_ID,$lietotajs ) // iesniegto mājasdarbu dzēšana
{
    $this->db->query("call delete_file($lietotajs,$md_faili_ID) ");
    $affected=$this->db->affected_rows();
    echo '1' ;
}

public function greaterDate($start_date,$end_date) //datumu salīdzināšanas funkcija
{
    $start = strtotime($start_date);
    $end = strtotime($end_date);
    if ($start-$end > 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

public function load_teacher_info($prieksmets_ID, $teacher_ID) // priekšmetu skolotāju kontaktu parādīšana
{
    $klases_ID=$this->session->userdata('klases_ID');
    $this->db->select('iela, pilseta, valsts, telefons, majas_nosaukums, pagasts, e_mail');
    $this->db->join('prieksmeti', 'registered_users.isn = prieksmeti.Skolotajs');
    $this->db->join('adrese', 'registered_users.adrese_ID_FK = adrese.adrese_ID');
}

```

```

    $query=$this->db->get_where('registered_users', array('klase_parb' => $klases_ID, 'prieks_ID'=>$prieksmets_ID,
'Skolotajs'=>$teacher_ID ));
    $items = $query->result();
    $a=sizeof($items) ;
    if($a>0)
    { // izdrukā skolotāja kontaktus (vajadzīgs ajax funkcijai)
        echo "Kontakti: <br/> E-pasts: ".$items[0]->e_mail."<br/>telefons: ".$items[0]->telefons.
            "<br/> ielas adrese: ".$items[0]->iela ."<br/>pilsēta: ".
            $items[0]->pilsēta."<br/> valsts: ".$items[0]->valsts."<br/>" ;
        if(isset($items[0]->pagasts))
        {
            echo "pagasts: ".$items[0]->pagasts ;
        }
        if(isset($items[0]->majas_nosaukums))
        {
            echo "mājas nosaukums: ".$items[0]->majas_nosaukums ;
        }
    }else
    {
        echo "Nav atrasti kontakti šim lietotājam!" ;
    }
}
?>

```

Bakalaura darbs “PHP IETVARU SALĪDZINĀJUMS UN IZVĒLE  
ATKARĪBĀ NO PROJEKTU MĒRĶIEM” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti.

Autors: **Krišjānis Kokorīts** \_\_\_\_\_

Rekomendēju darbu aizstāvēšanai

Vadītājs: **Dr. vadībzin. doc. Imants Gorbāns** \_\_\_\_\_

Darbs iesniegts Datorikas fakultātē

Dekānā pilnvarotā persona:

Darbs aizstāvēts bakalaura darba gala pārbaudījuma komisijas sēdē

\_\_\_\_\_ prot. Nr. \_\_\_\_

Komisija: