

Житомирський державний університет імені Івана Франка

Фізико-математичний факультет

Кафедра прикладної математики та інформатики

**ДИПЛОМНА РОБОТА**

з інформатики

на тему:

Використання фреймворку Yii2 для розробки автоматизованої системи  
збору даних

Студента 6 курсу 73 групи

напряму підготовки 6.040302 «Інформатика»

Гришка А. В.

Керівник: Кривонос О. М.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_

Члени комісії \_\_\_\_\_

(підпис) (прізвище та  
ініціали)

\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Житомир – 2017 рік

## ЗМІСТ

Вступ.....	3
РОЗДІЛ I. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО РОЗРОБКУ ПЗ .....	5
1.1. Розвиток програмування.....	5
1.2. Парадигми програмування.....	7
1.3. Мова програмування PHP .....	10
1.4. Стандартна бібліотека PHP.....	12
1.5. PHP 7 .....	13
1.6. Системи управління базами даних(СУБД) .....	14
1.7. Різновиди систем управління базами даних .....	17
РОЗДІЛ II. ПІДГОТОВКА ДО ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ .....	18
2.1. Фреймворк Yii .....	18
2.2. Коротка історія розвитку Yii .....	21
2.3. Встановлення Yii.....	22
2.3. PhpStorm.....	24
2.4. MySQL .....	28
РОЗДІЛ III. ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ЗБОРУ ДАНИХ .....	30
3.1. Підготовка проекту .....	30
3.2. Підготовка бази даних .....	35
3.3. Налаштування користувачів.....	38
3.4. Генерація заяв.....	40
3.5. Сторінка налаштувань.....	43
3.6. Генерація звітів.....	44
ВИСНОВКИ .....	47
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	48

## Вступ

Стрімкий розвиток комп'ютерної техніки та програмного забезпечення - одна з найвагоміших рис сучасного суспільства. Комп'ютерні технології проникли практично в усі сфери людської діяльності. Їх застосовують у навчальних закладах і великих бібліотеках, у парламентах і міністерствах, у банках і на складах, у системах зв'язку і системах управління транспортом, у податкових інспекціях і у сфері розваг тощо. Тепер практично неможливо уявити сучасний офіс без повсякденного і широкого застосування комп'ютерних технологій. Комп'ютер став неодмінним атрибутом багатьох професій.

Розробка програмного забезпечення вимагає від програмістів дотримання критеріїв якості та вимогам часу для розробки. Для спрощення процесу розробки створюються бібліотеки та фреймворки, які дозволяють покращити якість продукту(зазвичай такі рішення створюються і тестуються цілими спільнотами розробників), а також сприяють збільшенню швидкості розробки.

В цій роботі описується створення програмного забезпечення мовою PHP, з використанням вільно доступного фреймворку Yii2, з використанням інтегрованого середовища розробки PhpStorm.

**Актуальність** теми зумовлена тим, що в сучасних ринкових умовах важливим чинником є підвищення якості та ефективності програмного забезпечення. Сучасне програмне забезпечення має свої особливості і вимоги. Як і інші традиційні інженерні дисципліни, розробка програмного забезпечення має справу з проблемами якості, вартості та надійності. Багато програм містять мільйони рядків коду, які мають правильно виконуватись навіть при внесенні змін. Складність програмного забезпечення можна порівняти з складністю найбільш складних з сучасних машин, таких як літаки.

**Метою** даної роботи є аналіз практичних можливостей розробки автоматизованої системи збору даних з використанням Yii2.

**Предметом** дослідження даної роботи є можливості та інструменти для розробки програмного забезпечення.

**Об'єктом** дослідження даної роботи є особливості використання мови програмування PHP та фреймворку Yii2 для розробки програмного забезпечення.

Завданнями даної роботи є:

- ознайомитись з мовою PHP
- ознайомитись з IDE PhpStorm;
- ознайомитись з фреймворком Yii2;
- обрати інструменти для розробки;
- реалізувати автоматичну систему збору даних.

## РОЗДІЛ I. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО РОЗРОБКУ ПЗ

### *1.1. Розвиток програмування*

Розробка програмного забезпечення (англ. software engineering, software development) — це рід діяльності (професія) та процес, спрямований на створення та підтримку працездатності, якості та надійності програмного забезпечення, використовуючи технології, методологію та практики з інформатики, керування проектами, математики, інженерії та інших областей знання[4]. Бурний розвиток технологій програмування безпосередньо пов'язаний з швидким розвитком науково-технічного прогресу і комп'ютерної техніки зокрема.

Програми для перших обчислювальних машин створювались, як правило, в машинних кодах або на асемблері(англ. assembler)[5] і вважались витвором мистецтва, бо їм вистачало мініатюрного за сучасними поняттями об'єму пам'яті. Пошуки помилки в програмі можна було, мабуть, порівняти з муками Тантала. Програмісти були схожі на “вищу касту” серед нормальних людей, бо вони єдині були здатні на спілкування з обчислювальною технікою. Цей етап програмування називають “стихійним програмуванням”. Створення нових алгоритмічних мов програмування, таких як FORTRAN та ALGOL, не суттєво змінило корені ситуацію. Дані в програмах зберігались, як правило, в глобальних областях, які спільно використовувались різними підпрограмами.

Наступним етапом стало структурне програмування. До створення робочої групи з методології програмування при Міжнародній федерації по обробці інформації. До її складу увійшло багато відомих програмістів, наприклад, Н. Вірт, П. Наур, Ч.Хоар, У. Дал, Е. Дейкстра. Їх спільні зусилля привели до оформлення цієї нової технології (інколи кажуть – парадігми)[7,8]. Завдяки принципам структурного програмування вдалося подолати фактор складності та зрозуміти причини невдач програмних проектів великого масштабу. Цього вдається досягти завдяки правильній

структурі проекту в цілому і його модулів поодиноці. Створена у 1972 році мова програмування С (сі) є одною з найпопулярніших мов і донині.

Сучасні технології програмування базуються на принципах об'єктно-орієнтованого програмування, завдяки якому складні програмні проекти реалізуються у вигляді сукупності об'єктів певної ієрархії. Їх взаємодія встановлюється шляхом передачі повідомлень між об'єктами. На підтримку нової технології програмування були створені нові мови, наприклад С++(яка спочатку мала назву Сі з класами)[3], Java, С#. Організація програм на трьох китах - інкапсуляції, успадкування, поліморфізму дозволила значно підвищити рівень програмних проектів.

## ***1.2. Парадигми програмування***

### **Процедурне програмування**

Першою відомою парадигмою було *процедурне програмування*. Основна думка такого підходу: виявіть які процедури вам потрібні, опишіть їх в програмі та використовуйте найкращі з відомих алгоритмів.

Основні наголоси робились на обробку даних за допомогою алгоритму, котрий проводив потрібні обчислення. Для підтримки цього способу, різні мови надавали механізм передачі параметрів та отримання результатів функцій. Першою. процедурною мовою був Фортран - імперативна мова програмування загального призначення, яка особливо підходить для інтенсивних чисельних та наукових обчислень [13]. Такі мови як Алгол, Паскаль та С продовжили розвивати цей напрям.

### ***Модульне програмування***

З часом при проектуванні програм вимоги змістились з організації процедур до організації структур даних. Це також було зумовлено зростанням розмірів програм. Модулем зазвичай називають сукупність зв'язаних процедур і даних, котрими вони маніпулюють. Парадигма мала вид: виявіть які модулі вам потрібні; поділіть програму так, щоб дані були приховані в цих модулях.

Ця парадигма відома також як «Принцип приховування даних». Якщо в мові немає можливості згрупувати процедури разом за даними, то він погано підтримує модульний стиль програмування. Оскільки дані єдина річ, яку хочуть приховати, поняття приховування даних розширюється до поняття приховування інформації – імен змінних, констант, функцій та типів.

### ***Абстракція даних***

Кожен модуль, котрий керує типом, повинен визначати свій власний алгоритм створення змінних цього типу. Не існує універсальних правил присвоєння ідентифікаторів, котрі можуть визначати об'єкти цього типу. У «змінних» таких типів не існує імен, котрі були б відомі транслятору або

іншим системним програмам, ці змінні не підкоряються звичайним правилам областей видимості та передачі параметрів.

Тип, котрий реалізується управляючим ним модулем, по багатьом аспектам відрізняється від стандартних типів. Такі типи не отримують того ж контролю від транслятор, котрий забезпечується для вбудованих типів. В таких мовах як PHP, C++, Java, C# та подібним, дозволяється створювати власні типи, котрі трактуються практично так як вбудовані. Такі типи зазвичай називають абстрактними, або користувацькими. Парадигму можна виразити – підготуйте потрібні типи, та опишіть повний набір операцій для кожного. Більшість модулів доречно створювати абстрактними типами.

### ***Об'єктно-орієнтоване програмування***

Об'єктно-орієнтоване програмування(ООП) найкращим чином представляє технологію управління елементами будь-якої складності, створюючи умови для багатократного використання програмних компонентів та об'єднання даних з методами їх обробки. Суть ООП полягає в використанні концепції об'єктів. Ідея підходу полягає в зв'язуванні даних з методами, що ці дані оброблюють в єдине ціле – об'єкт.

Фактично ООП можна розглядати як модульне програмування нового рівня, коли замість механічного об'єднання процедур і даних, акцент робиться на їх осмислений зв'язок.

Об'єктна модель спроможна однаково добре описати як елементи управління графічного інтерфейсу(GUI), так і реальні об'єкти(автомобіль, тварина). Таким чином завдання ООП складається в те, щоб правильно представити ці об'єкти на мові програмування.

Мови програмування котрі мають підтримку ООП повинні включати в себе три кита цього способу:

- інкапсуляція – полягає в тому, що об'єкт містить не лише дані, але й методи, котрі містять правила їх обробки;
- наслідування – створення нових класів, котрі наслідують дані та методи одного чи кількох раніше створених типів. При цьому можливе



додавання нових методів та перезапис існуючих. В результаті створюється так звана ієрархія класів;

- поліморфізм – присвоєння методу єдиного імені або ідентифікатора в рамках ієрархії класів, таким чином, щоб будь-який клас в ієрархії мав можливість по своєму виконувати зв'язані з цим методом дії.

З появою цієї парадигми та деталізацією концепції почали з'являтися нові мови програмування, де ООП лягло в основу. Серед них Smalltalk, C++, Java, C#, PHP.

### *1.3. Мова програмування PHP*

PHP (англ. PHP: Hypertext Preprocessor — PHP: гіпертекстовий препроцесор), попередня назва: Personal Home Page Tools — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення[2].

PHP транслюється веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скрипкових мов, таких як JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевага з точки зору безпеки. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

PHP — мова, код якої можна вбудовувати безпосередньо в HTML-код сторінок, які, у свою чергу, будуть коректно оброблені PHP-інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (`<?php`) і продовжує виконання до того моменту, поки не зустрине закриваючий тег (`?>`).

Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.

В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase. Через стандарт відкритого інтерфейсу зв'язку з базами даних (Open Database Connectivity Standard — ODBC) можна підключатися до всіх баз даних, для яких існує драйвер.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl і C і спеціально спрямована на роботу в Інтернеті, мова

з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що в наш час є мало не найпопулярнішою мовою для створення веб-застосунків (скриптів).

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, безсумнівно справили благотворний вплив на багато проектів, в першу чергу — Linux хоч і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту PHP.

Ухвалення стратегії Open Source і безплатне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам. Окрім цього, користувачі PHP в усьому світі є свого роду колективною службою підтримки, і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків.

## 1.4. Стандартна бібліотека PHP

Стандартна бібліотека PHP (SPL) – це набір інтерфейсів і класів, для вирішення стандартних завдань[3].

Не потребує ніяких зовнішніх бібліотек для збірки рішення, доступно за замовчуванням в версії PHP 5.0.0 та вище.

SPL включає в себе ряд стандартних структур даних, літераторів для проходження об'єктів, інтерфейсів, стандартних виключень, декілька класів для роботи з файлами та інші функції, наприклад `spl_autoload_register()`.

З версії 5.3 в PHP були введені простори імен. Всі стандартні класи та інтерфейси знаходяться в глобальному просторі[3].

Деякі структури даних:

- `SplDoublyLinkedList` - двох-зв'язний список;
- `SplHeap` - купа;
- `SplPriorityQueue` – черга з пріоритетом;
- `SplQueue` - черга перший прийшов – перший пішов;
- `SplStack` - стек перший прийшов – останній пішов.

Ітератори:

- `DirectoryIterator` - переглядає каталоги файлової системи;
- `FilesystemIterator` - ітератор файлової системи;
- `RecursiveArrayIterator` – рекурсивний ітератор.

Виключення:

- `LogicException` – використовується для помилок в логіці програми;
- `LengthException` – помилка в довжині;
- `RangeException` – невірний інтервал значень;
- `OverflowException` - використовується при доданні елемента в заповнений контейнер.

## 1.5. PHP 7

3 грудня 2015 року було випущено першу стабільну нову версію мови PHP 7[2].

Нова версія базується на експериментальній гілці phpng (PHP Next Generation – наступне покоління), та розроблювалась задля збільшення швидкості виконання і зменшення використовуваної пам'яті.

Найголовніші та найбільш використовувані нововведення PHP 7:

- Скалярні типи параметрів та результатів функцій;
- Spaceship оператор(<=>) – повертає 0 якщо операнди рівні, -1 якщо перший менше, 1 якщо більше;
- Оператор ?? – повертає значення за замовчуванням, якщо змінна дорівнює null;
- Анонімні класи;
- Поліпшення підтримки 64 розрядних систем;
- Обробка фатальних помилок.

В даний момент активно іде робота над виправлення помилок.

1 грудня 2016 року було випущено PHP 7.1[18].

Нова версія містить в собі виправлення існуючих помилок інтерпретатора та новий функціонал.

Найголовніші та найбільш використовувані нововведення PHP 7.1:

- Функція може повертати тип void, варто зазначити що null не є void;
- Можливість використовувати обнуляемі типи(типи які мають скалярне значення чи null);
- Від'ємні індекси доступу до рядків;
- Конвертація callable в замикання;
- Підтримка областей видимості констант;
- Перехоплення декількох виключних ситуацій в одному блоці.

В даний момент активно іде робота над новим функціоналом та виправлення помилок.

### *1.6. Системи управління базами даних(СУБД)*

База даних (БД) - це сукупність взаємозв'язаних даних, що зберігаються разом[19]. Основними та невід'ємними властивостями БД є такі:

- для даних допускається така мінімальна надлишковість, яка сприяє їх оптимальному використанню в одному чи кількох застосуваннях;
- незалежність даних від програм;
- для пошуку та модифікації даних використовуються спільні механізми;
- як правило, у складі БД існують засоби для підтримки її цілісності та захисту від неавторизованого доступу;

На відміну від файлових систем БД зорієнтована для підтримки даних для кількох застосувань. На практиці ця властивість інколи порушується.

Взаємозв'язаність даних полягає в тому, що доступ до певної групи даних якогось застосування загалом полегшує доступ до інших груп даних цього ж застосування. В умовах орієнтації БД на велику кількість застосувань виникає необхідність у підтримці значного числа різноманітних зв'язків між даними. Саме у розумінні тісного логічного зв'язку використані слова про збереження разом даних.

База даних – це набір даних з наступними властивостями:

- дані логічно пов'язані між собою і несуть відповідну інформацію;
- структура баз даних звичайно відповідає тому специфічному набору даних, які вона містить;
- бази даних відображають тільки окремі аспекти реального світу, що дає змогу визначити їх як "мікросвіт".

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні

елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші.

Необхідність програмувати завжди утримувала устаткування баз даних в малому бізнесі. Великі підприємства могли дозволити собі зробити наказ на програмування спеціальної системи „під себе“. Малим підприємствам звичайно не по силам було не тільки вирішити, але й правильно сформулювати цю задачу.

Становище покращилось з появою у складі пакета Microsoft Office системи управління базами даних Access. Ранні версії цієї програми мали номери Access 2.0 та Access 95.

За допомогою Access 9x звичайні користувачі отримали зручний засіб для створення та експлуатації досить міцних баз даних без необхідності щось програмувати. У той же час робота з Access 9x не викреслює можливості програмування. За бажанням систему можна розвивати та налаштовувати своїми силами. Для цього треба володіти основами програмування на мові Basic.

Ще одним додатковим досягненням Access 9x є інтегрованість цієї програми з Excel 9x, Word 9x та іншими програмами пакета Office 9x. Данні, створені в різних додатках, утворюючих цей пакет, легко імпортуються та експортуються з одного додатка в інший.

Системи управління базами даних (СУБД) є набором програмних засобів, необхідних для створення, використання і підтримки баз даних.

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

Організація єдиної бази даних стала можливою лише завдяки тому, що були створені спеціальні програмні продукти — системи управління базами даних (СУБД).

Основне призначення СУБД — створення та підтримка в актуальному стані бази даних, а також зв'язок її з програмами розв'язування економічних завдань (прикладні програми користувачів).

База даних – це комп'ютерний термін, який використовується для позначення сукупності інформації з окремої теми або відомостей, пов'язаних з деякою прикладною задачею. Зберігання інформації у вигляді бази даних полегшує доступ до неї, пошук та вилучення потрібних фрагментів.

На магнітному диску база даних може зберігатись у вигляді одного файла (бази даних MS Access, Informix та ін.) або у вигляді папки з файлами (бази даних Interbase, Paradox та ін.).



## ***1.7. Різновиди систем управління базами даних***

### **Системи управління даними першого покоління**

СУБД першого покоління характерні тим, що кожна група користувачів розробляла своє власне програмне забезпечення по управлінню даними. Наслідками такої сепаратизації стало надмірне дублювання програмних кодів і даних.

### **Системи управління даними другого покоління**

Файли взаємопов'язаних даних об'єднуються в бази даних. СУБД створюються для таких досвідчених користувачів, як програмісти.

### **Системи управління даними третього покоління**

Можливості СУБД розширились. Створені розвинуті інтерфейси, що забезпечують інтерактивний доступ звичайним користувачам.

Переваги СУБД :

- Скорочення надлишку даних;
- Без баз даних неможливо уникнути зберігання надлишкових даних;
- При наявності центрального контролю баз даних деякі надлишкові дані можна усунути;

Надлишкові дані не можуть бути повністю усунені, оскільки велику роль в СУБД відіграють питання часу і достовірності.

У світі існує безліч СУБД. Незважаючи на те, що вони можуть по-різному працювати з різними об'єктами і надають користувачу різні функції й засоби, більшість СУБД спираються на єдиний устояний комплекс основних понять. Це дає нам можливість розглянути одну систему й узагальнити її поняття, прийоми й методи на весь клас СУБД.

Найпоширенішими СУБД є MySQL, PostgreSQL та Microsoft Access.

## РОЗДІЛ II. ПІДГОТОВКА ДО ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ

### 2.1. Фреймворк Yii



Yii (вимовляється як «І» або [ji:]) — це сучасний веб-фреймворк, написаний на мові програмування PHP та реалізує парадигму модель-вид-контролер(model-view-controller, MVC). Yii китайською означає «простий та елегантний», також може розшифруватись як «Yes It Is!»[4].

Yii – це універсальний фреймворк і може бути використаний у багатьох веб-додатках, що використовують PHP. Завдяки його компонентній структурі і відмінній підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів як портали, форуми, системи керування вмістом (CMS), інтернет-магазини або RESTful-додатки.

Якщо ви вже знайомі з іншими фреймворками, вам напевно буде цікаво порівняти їх із Yii:

- Як і багато інших PHP-фреймворків, Yii втілює архітектурний шаблон MVC (Model-View-Controller) та сприяє організації коду відповідно до вимог шаблону;
- Yii дотримується філософії простого й елегантного коду. Yii ніколи не буде намагатись пере-ускладнювати дизайн тільки заради слідування будь-яким шаблонам проектування;
- Yii є full-stack фреймворком і включає в себе перевірені можливості, які добре себе зарекомендували: конструктори запитів та ActiveRecord для реляційних та NoSQL баз даних, підтримка REST API, багаторівневе кешування та інші;
- Yii надзвичайно розширюваний. Ви можете налаштувати або замінити практично будь-яку частину основного коду. Завдяки надійній архітектурі розширень Yii, досить легко використовувати або розробляти поширювані розширення;

- Висока швидкодія завжди є головною ціллю Yii.

Yii — не проект однієї людини. Він підтримується і розвивається сильною командою і великою спільнотою розробників, які їй допомагають. Команда розробників фреймворку Yii стежать за тенденціями веб-розробки і розвитком інших проектів. Найбільш значимі можливості та кращі практики регулярно впроваджуються у фреймворк у вигляді простих й елегантних інтерфейсів.

Основні можливості:

- Висока продуктивність;
- Паттерн Модель-вид-контролер;
- Інтерфейси DAO та Active Record для роботи з базами даних (PDO);
- Підтримка інтернаціоналізації;
- Кешування сторінок та окремих фрагментів;
- Перехоплення та обробка помилок;
- Введення та валідація веб-форм;
- Автентифікація та авторизація;
- Використання AJAX та інтеграція з jQuery;
- Генерація базового PHP-коду для CRUD-операцій (скаффолдінг);
- Підтримка тем оформлення для їх легкої зміни;
- Можливість підключення сторонніх бібліотек;
- Міграції бази даних;
- Автоматизоване тестування;
- Підтримка REST.

На даний момент існує дві основні версії Yii: 1.1 та 2.0. Версія 1.1 є попереднім поколінням і знаходиться у стані підтримки. Версія 2.0 - це повністю переписаний Yii, що використовує останні технології і протоколи, такі як Composer, PSR, простори імен, трейти і багато іншого. 2.0 - поточне покоління фреймворку. На цій версії будуть зосереджені основні зусилля

кілька наступних років. Даний посібник призначений в основному для версії 2.0.

Yii 2.0 потребує PHP 5.4.0 та вище. Щоб дізнатися вимоги для окремих можливостей ви можете запустити скрипт перевірки вимог, який поставляється із кожним релізом фреймворку.

Для розробки на Yii необхідне загальне розуміння ООП, оскільки фреймворк повністю слідує цій парадигмі. Також слід вивчити такі сучасні можливості PHP як простори імен і трейти.

На даний момент фреймворк сумісний з PHP 7.

## *2.2. Коротка історія розвитку Yii*

Історія Yii почалася 1 січня 2008 року, як проект з виправлення деяких вад фреймворку PRADO (PHP Rapid Application Development Object-oriented), що став в 2004 переможцем «Zend PHP 5 coding contest»[4].

Фреймворк PRADO був спробою перенести елементи ASP.NET на платформу PHP, включаючи такі основні компоненти як ViewState, PostBacks, Page\_Load та OnClick. Якщо подивитися вихідний код фреймворку, то можна побачити що деякі шматки коду просто запозичені з класичного ASP.NET (наприклад, поділ на Rare Fields та Occasional Fields в класі Control[4] з метою оптимізації використання пам'яті, яка має сенс в .NET, але являє сумнівну цінність в PHP).

PRADO успадкував від ASP.NET майже всі негативні сторони: повільно обробляв складні сторінки, мав круту криву навчання і був достатньо важкий у налаштуванні.[4]

У певний момент автор (Qiang Xue) зрозумів, що PHP-фреймворк повинен бути побудований дещо по-іншому, і ось в жовтні 2008 року, після більш 10 місяців закритої розробки, вийшла перша альфа-версія. 3 грудня 2008 був випущений Yii 1.0.

12 жовтня 2014 року був випущений Yii 2.0. Не маючи зворотної сумісності з першою версією Yii2 був створений з використанням сучасних технологій та бачень.

### 2.3. Встановлення Yii

Yii використовує менеджер пакунків Composer. Також можлива інсталяція за допомогою архіву з офіційного сайту, але це застарілий та не рекомендований спосіб. Composer дозволяє швидко встановлювати потрібні додатки до проекту, тому він активно використовується спільнотою. Не має графічного інтерфейсу, лише консоль.

Після стандартного встановлення Yii ми отримуємо як фреймворк, так і шаблон проекту. Шаблон проекту - це робочий проект Yii, в якому реалізовано деякий базовий функціонал, такий як система входу/виходу користувачів, форма зворотнього зв'язку і т. д. Його код організовано в рекомендований спосіб. Таким чином, це може слугувати гарною відправною точкою для ваших проектів.

Базовий шаблон проекту підходить для розробки 90 відсотків веб-додатків. Він відрізняється від Розширеного шаблону проекту в основному організацією коду.

Якщо у вас все ще не встановлено Composer, то це можна зробити за допомогою інструкції на [getcomposer.org](http://getcomposer.org). При роботі з Windows, необхідно завантажити та запустити `Composer-Setup.exe`.

Якщо ж Composer вже було встановлено раніше, переконайтесь, що використовуєте його останню версію. Ви можете оновити Composer простою командою *`composer self-update`*.

Після встановлення Composer, встановити Yii можна виконавши наступну команду з директорії, яка доступна через Web:

```
composer global require "fxp/composer-asset-plugin:^1.2.0"  
composer create-project --prefer-dist yiisoft/yii2-app-basic basic
```

Перша команда встановить плагін ресурсів composer (`composer-asset-plugin`), що дозволить керувати залежностями пакунків Bower та NPM за допомогою Composer. Цю команду потрібно виконати лише один раз. Друга команда встановить Yii у директорію під назвою `basic`. За бажанням, ви можете обрати інше ім'я для директорії.

Під час встановлення може статися так, що Composer запитає облікові дані від вашого профілю на GitHub, через встановлені обмеження запитів GitHub API. Це є нормальним, оскільки Composer повинен отримати багато інформації для всіх пакунків із GitHub. Надання облікових даних профілю GitHub збільшить кількість запитів до API, потрібних для подальшої роботи Composer. Для більш детальної інформації, будь ласка, зверніться до документації Composer.

Варто зауважити, що нестабільну версію Yii не можна використовувати на робочому сервері, оскільки вона може порушити виконання робочого коду.

Після успішного встановлення ви можете налаштувати свій веб-сервер (див. наступний розділ) або використати вбудований веб-сервер PHP, виконавши наступну консольну команду із директорії web:

```
php yii serve
```

За замовчуванням, HTTP-server буде прослуховувати порт 8080. Проте, якщо цей порт вже використовується, або ви бажаєте таким чином використовувати кілька додатків одразу - ви можете встановити, який саме порт використовувати - додайте аргумент `--port`:

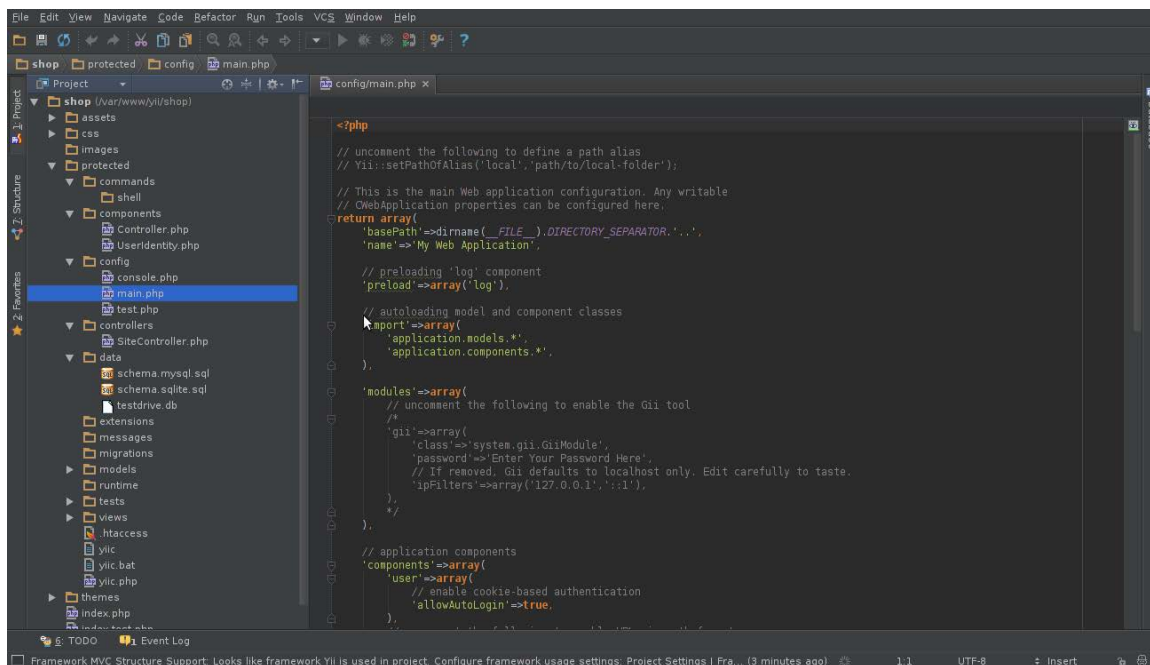
```
php yii serve --port=8888
```

Тепер ви можете використати свій браузер для доступу до встановленого Yii додатку за наступним посиланням:

```
http://localhost:8080/
```

## 2.3. PhpStorm

JetBrains PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA[15].



PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3 – 7 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

Вся функціональність WebStorm включена в PhpStorm.



## *Особливості*

Основна задача IDE — спростити розробку програми з допомогою PHP на різних платформах. Тому серед можливостей, які властиві будь-якому середовищу розробки, є і специфічні.

### *Редактор коду PHP*

PhpStorm надає багатий і інтелектуальний редактор коду для PHP з підсвічуванням коду, розширеною конфігурацією форматування коду, перевіркою на наявність помилок на льоту і розумним автодоповненням. [5]

- Підтримка PHP 5.3, 5.4 та 5.5, включаючи генератори, співпрограми, простори імен, замикання, типажі, синтаксис коротких масивів, доступ до члена класу при інстанціюванні, розіменування масиву при виклику функції, бінарні літерали, вираження в статичних виклики тощо. PhpStorm може використовуватися як для сучасних, так і для традиційних проектів на PHP;
- Автодоповнення коду фіналізують класи, методи, імена змінних, ключові слова PHP, а також широко використовувані імена полів і змінних залежно від їхнього типу;
- Підтримка стандартів оформлення коду (PSR1/PSR2, Drupal, Symfony2, Zend);
- Підтримка PHPDoc. PhpStorm надає відповідне автодоповнення коду, засноване на анотаціях @property, @method і @var.
- Детектор дубльованого коду;
- PHP Code Sniffer (phpcs), котрий перевіряє код на льоту;
- Рефакторинги (перейменування, введення змінної/константи/поля, вбудовування змінної);
- Підтримка редагування шаблонів Smarty (підсвічування синтаксичних помилок, автодоповнення функцій і атрибутів Smarty, автоматична вставка парних дужок, лапок і закриваючих тегів тощо);
- MVC подання для фреймворків Symfony2 і Yii;
- Розпізнавання коду, запакованого в PHAR-архіві.

### *Середовище розробки*

- Підтримка SQL і баз даних ( Рефакторинг схеми бази даних, генерація скриптів міграції схеми, експорт результатів виконання запиту у файл або буфер обміну, редагування збережених процедур і багато іншого);
- Віддалене розгортання додатків і автоматична синхронізація з використанням FTP , SFTP , FTPS та ін протоколів;
- Інтеграція з системами управління версіями ( Git - включаючи спеціальний функціонал для роботи з GitHub , Subversion , Mercurial , Perforce , CVS , TFS ), що дозволяє робити багато дій, наприклад commit, merge, diff та інші, прямо з PhpStorm;
- Локальна історія (Local History) (локально відстежує будь-які зміни в коді);
- PHP UML (Діаграми класів UML для PHP коду з рефакторингом, що викликаються прямо з діаграми);
- Підтримка Phing (надає автодоповнення, перевірку стандартних тегів, властивостей, імен цілей, значень атрибутів шляху в компоновальних файлах (build files));
- Інтеграція з системами відстеження помилок;
- Підтримка Vagrant, SSH консолі і віддалених інструментів;
- Підтримка Google App Engine For PHP.

### *Зневадження і тестування*

- Легко конфігурований візуальний зневаджувач (Xdebug, Zend Debugger) для перевірки відповідних контексту локальних змінних і заданих користувачем об'єктів стеження, у тому числі масивів і складних об'єктів, а також редагування значень на льоту;
- Інтеграція з профілювальником : скрипти можна профілювати прямо з PhpStorm за допомогою Xdebug або Zend Debugger. Доступний агрегований звіт; користувач може перейти від статистики виконання прямо до функції в PHP коді;

- Інтеграція з фреймворком модульного тестування PHPUnit (тести PHPUnit можна розробляти в PhpStorm і відразу запускати з директорії, файлу або класу за допомогою контекстного меню) з покриттям коду (code coverage).

### ***Робота з JavaScript, CSS і HTML***

- Вся функціональність, доступна в WebStorm, включена в PhpStorm;
- Автодоповнення коду для JavaScript, HTML і CSS (для тегів, ключових слів, міток, змінних, параметрів і функцій);
- Підтримка HTML5;
- Live Edit: зміни в коді можна миттєво переглянути в браузері без перезавантаження сторінки;
- Підтримка CSS/SASS/SCSS/LESS (автодоповнення коду, підсвічування помилок, валідація тощо);
- Zen Coding;
- Навігація по коду і пошук використань (перейти до оголошення / ідентифікатора, знайти використання);
- Підтримка ECMAScript Harmony;
- Рефакторинг для JavaScript (перейменування, виділення змінної / функції, вбудовування змінної / функції, переміщення / копіювання, безпечне вилучення, витяг вбудованого скрипта в окремий файл);
- Зневаджувач JavaScript, а також інтеграція з фреймворками модульного тестування JavaScript.

## 2.4. MySQL

Для практичної реалізації даної роботи необхідна база даних (БД). Серед готових рішень була обрана MySQL – потужна реляційна базу даних.

MySQL — вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування[16].

MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GP[16].

### *phpMyAdmin*

phpMyAdmin — веб-застосунок з відкритим кодом, написаний на мові PHP, представляє собою веб-інтерфейс для адміністрування СКБД MySQL. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць й баз даних. Додаток користується великою популярністю у веб-розробників, оскільки дозволяє керувати СКБД MySQL без безпосереднього вводу SQL команд, надаючи дружній інтерфейс[17].

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей додаток як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Програма розповсюджується під ліцензією GNU (General Public License), і тому деякі інші розробники інтегрують його у свої розробки, наприклад XAMPP, Denwer.

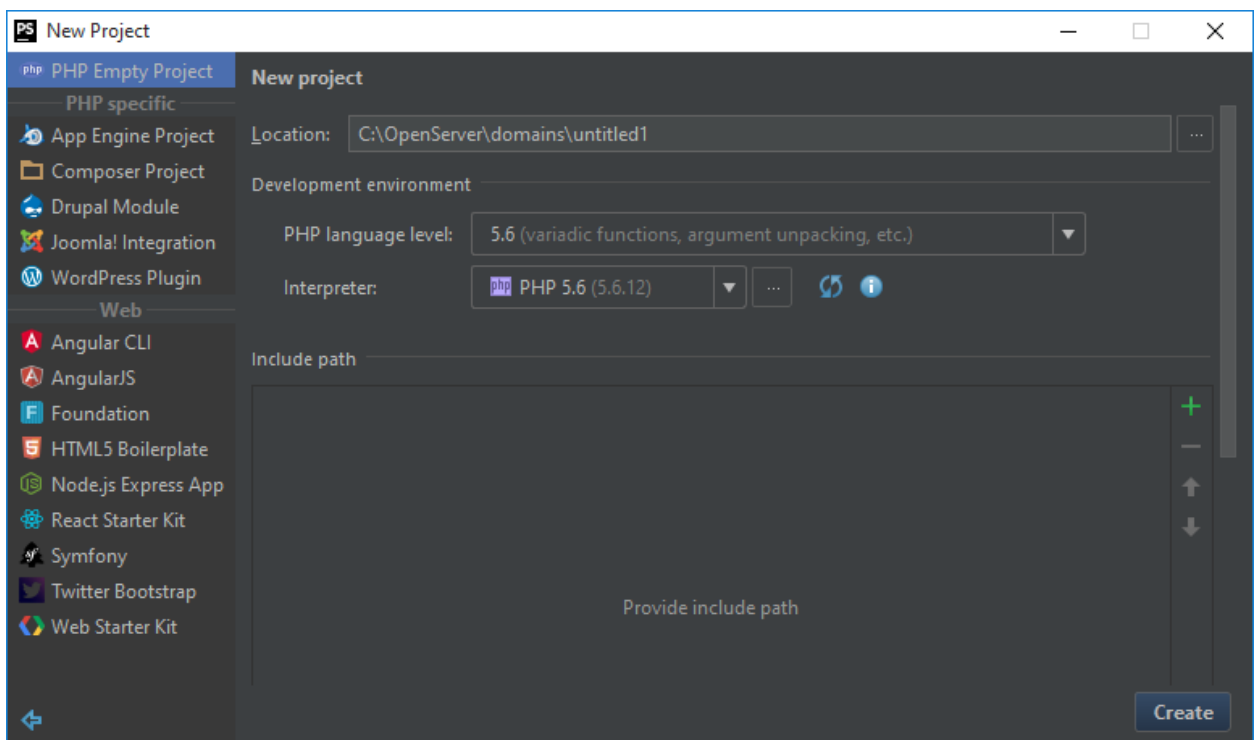
## РОЗДІЛ III. ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ЗБОРУ ДАНИХ

### 3.1. Підготовка проекту

Для проекту були обрані наступні технології:

- Мова програмування PHP - версія 5.6;
- СУБД MySQL 5.5;
- IDE JetBrains PhpStorm 2016.1;
- Система контролю версій Git;
- Веб хостинг для git - Github;
- Бібліотека Twitter Bootstrap;
- Бібліотека jQuery;
- OpenServer.

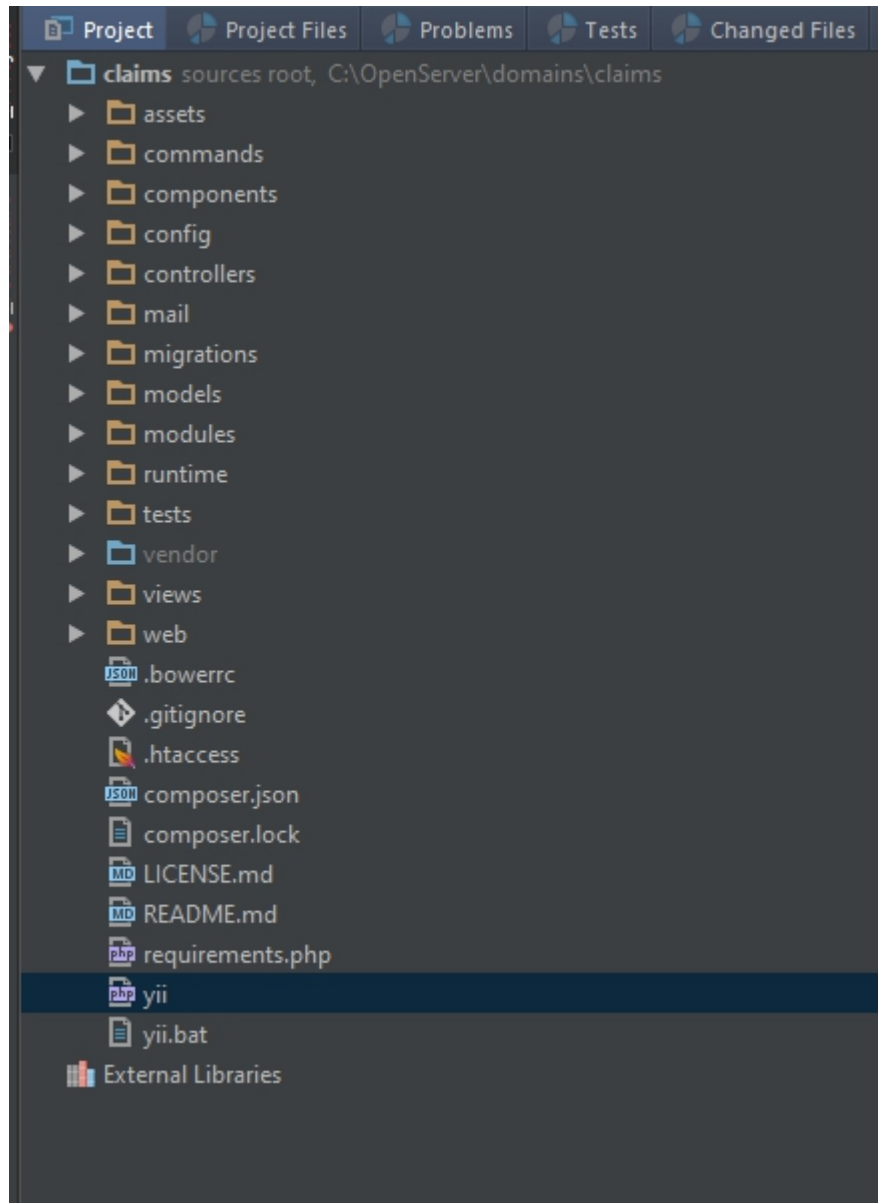
В PhpStorm було створено новий проект(вікно створення нового проекту показано на рисунку):



За допомогою менеджера пакетів Composer був встановлений фреймворк з стандартними налаштуваннями:

*composer create-project --prefer-dist --stability=dev yiisoft/yii2-app-basic*  
*claim*

Структура проекту має наступний вигляд:



- assets – налаштування використовуваних js та css файлів проекту;
- commands – консольні команди проекту;
- components – власні класи проекту;
- config – файли конфігурації;
- controllers – контролери для обробки запитів;
- mail – шаблони для відправки поштових листів;
- migrations – міграції бази даних;

- models – моделі даних;
- modules – модулі проекту(в даному проекті використовувався модуль адміністративної частини);
- runtime – кеш та логи проекту під час використання;
- tests – тести проекту;
- vendor – містить в собі залежності проекту(бібліотеки які використовуються в розробці);
- views – шаблони відображення;
- web – публічна директорія проекту при доступі з web мережі.

Слід зазначити, що в проекті ми будемо використовувати зрозумілі URL адреси(адреса яку може зрозуміти користувач, наприклад /products/view/1, замість index.php?module=products&action=view&id=1).

Конфігурація виглядає наступним чином:

```
<?php

$params = require(__DIR__ . '/params.php');

$config = [
    'id' => 'basic',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'language' => 'uk_UA', // мова проекту
    'modules' => [ // модулі
        'settings' => [
            'class' => 'app\modules\settings\Module',
        ],
    ],
    'components' => [
        'request' => [
            // !!! insert a secret key in the following (if it is empty) -
            // this is required by cookie validation
            'cookieValidationKey' => 'KfM7zGGyGFlrwFSxgDhqIzmtCzrs0ds5',
            'baseUrl' => '',
        ],
    ],
];
```



```

'authManager' => [ // налаштування для системи ролей користувачів
    'class' => 'yii\rbac\DbManager',
],
'cache' => [
    'class' => 'yii\caching\FileCache',
],
'user' => [
    'identityClass' => 'app\models\User', // клас користувача
    'enableAutoLogin' => true,
],
'errorHandler' => [
    'errorAction' => 'site/error', // стандартна дія для обробки
помилки при запиті
],
'mailer' => [
    'class' => 'yii\swiftmailer\Mailer',
    // send all mails to a file by default. You have to set
    // 'useFileTransport' to false and configure a transport
    // for the mailer to send real emails.
    'useFileTransport' => true,
],
'log' => [
    'traceLevel' => YII_DEBUG ? 3 : 0,
    'targets' => [
        [
            'class' => 'yii\log\FileTarget',
            'levels' => ['error', 'warning'],
        ],
    ],
],
'db' => require(__DIR__ . '/db.php'), // налаштування бази даних

'urlManager' => [
    'enablePrettyUrl' => true,
    'showScriptName' => false,
    'rules' => [
        'login' => 'site/login',
        'logout' => 'site/logout',
        'my-profile' => 'account/my-profile',
    ],
],

```

```

        ],
    ],
    'params' => $params,
];

if (YII_ENV_DEV) { // налаштування для debug режиму
    // configuration adjustments for 'dev' environment
    $config['bootstrap'][] = 'debug';
    $config['modules']['debug'] = [
        'class' => 'yii\debug\Module',
    ];

    $config['bootstrap'][] = 'gii';
    $config['modules']['gii'] = [
        'class' => 'yii\gii\Module',
    ];
}

return $config;

```

В свою чергу db.php виглядає наступним чином(використовується бібліотека PDO для роботи з базою даних):

```

<?php

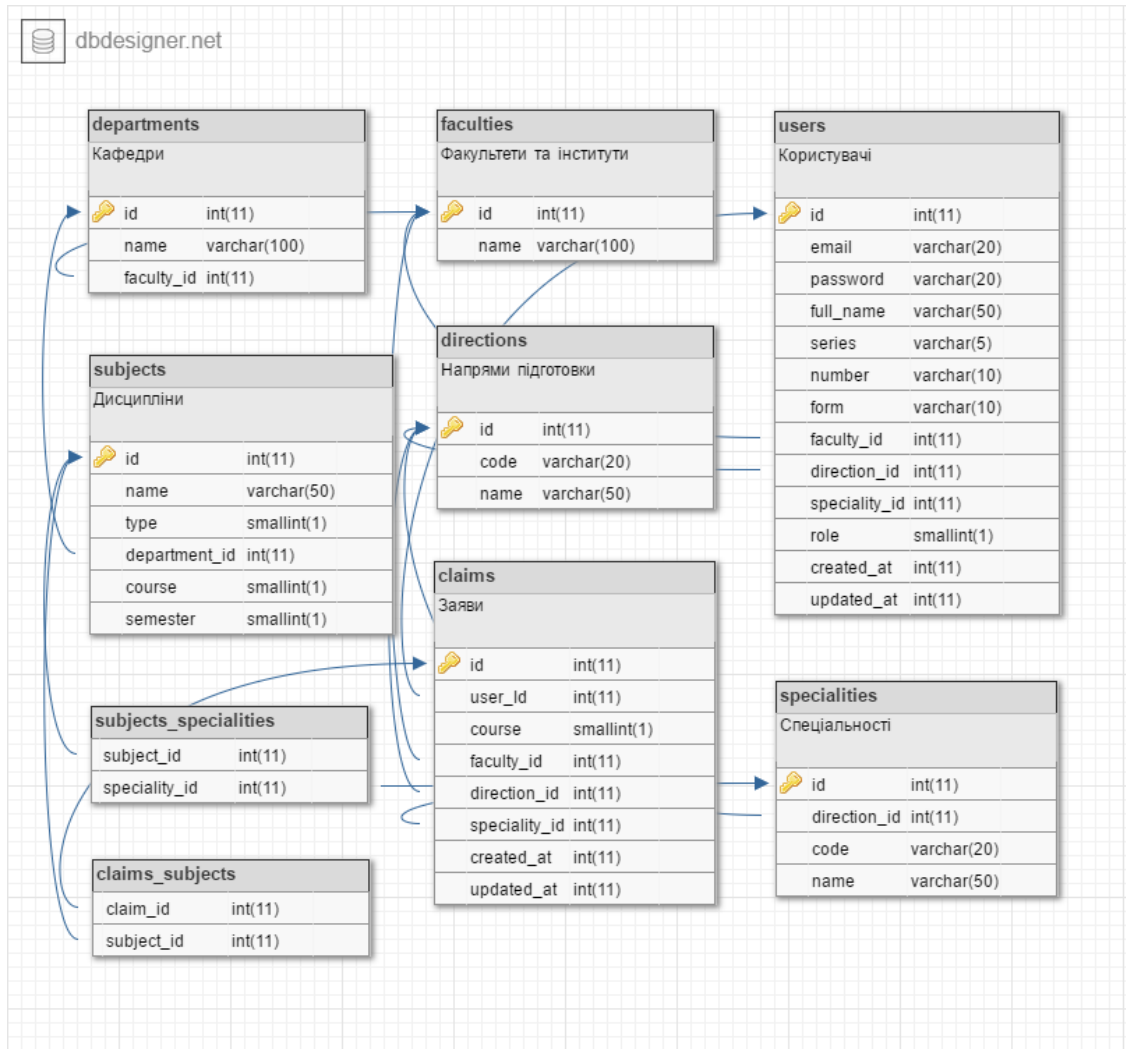
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=supportmad_claim',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
];

```

Файл params.php повертає порожній масив, бо в даному проєкті не було потреби в додаткових параметрах.

### 3.2. Підготовка бази даних

На рисунку нижче показана початкова схема таблиць бази даних для системи з показаними зв'язками між таблицями. Як і зазвичай є в процесі розробки до бази даних були додані додаткові поля.



Для практичної реалізації бази даних була використана MySql 5.5.45 та phpMyAdmin 4.0.10.

Для створення структури та заповнення початковою інформацією ми будемо використовувати міграції. Міграція – це своєрідна система контролю версій для бази даних. Yii2 підтримує механізм міграцій, має консольний додаток для роботи з ними, а також допоміжні методи створення таблиць, індексів та зв'язків. Таким чином ми можемо уникати написання SQL, який буде згенерований фреймворком.

Створити міграцію ми можемо за допомогою команди:

```
yii migrate/create migration_name
```

Ця консольна команда створить новий клас міграції *mYYMMDD\_HHMMSS\_migration\_name.php* (де YYMMDD\_HHMMSS – дата та час створення міграції в форматі UTC) в папці migrations(це стандартний каталог для зберігання міграцій, він може бути змінений в налаштуваннях).

Приклад файлу міграції використовуючи допоміжні методи фреймворку:

```
<?php
use yii\db\Migration;

class m160303_180536_add_subjects_table extends Migration
{
    public function safeUp()
    {
        $tableOptions = null;
        if ($this->db->driverName === 'mysql') {
            $tableOptions = 'CHARACTER SET utf8 COLLATE utf8_unicode_ci ENGINE=InnoDB';
        }

        $this->createTable('{{%subjects}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(255)->notNull()->unique(),
            'type' => $this->smallInteger(1)->notNull(),
            'department_id' => $this->integer(11)->notNull(),
            'course' => $this->smallInteger()->notNull(),
            'semester' => $this->smallInteger()->notNull(),
            'status' => $this->smallInteger(1)->notNull(),
        ], $tableOptions);

        $this->createIndex('subjects_search', '{{%subjects}}', ['type', 'course', 'semester', 'status']);
        $this->addForeignKey('subjects_departments', '{{%subjects}}', 'department_id', '{{%departments}}', 'id',
            'CASCADE');
    }

    public function safeDown()
    {
        $this->dropForeignKey('subjects_departments', '{{%subjects}}');
        $this->dropTable('{{%subjects}}');
    }
}
```

Метод `safeUp` використовується для застосування міграції до бази даних, там міститься дії які мають бути застосовані до бази даних, префікс `safe` говорить про те що команди будуть виконані з використанням режиму транзакції(якщо СУБД не підтримує транзакції то фреймворк видасть `Exception`).

Метод `safeDown` навпаки викликається при відміні міграції. Зазвичай там видаляються додані об'єкти. Метод також використовує транзакції.

Слід зазначити що для невеликих змін є методи `up` та `down`, які виконуються без транзакцій.

### 3.3. Налаштування користувачів

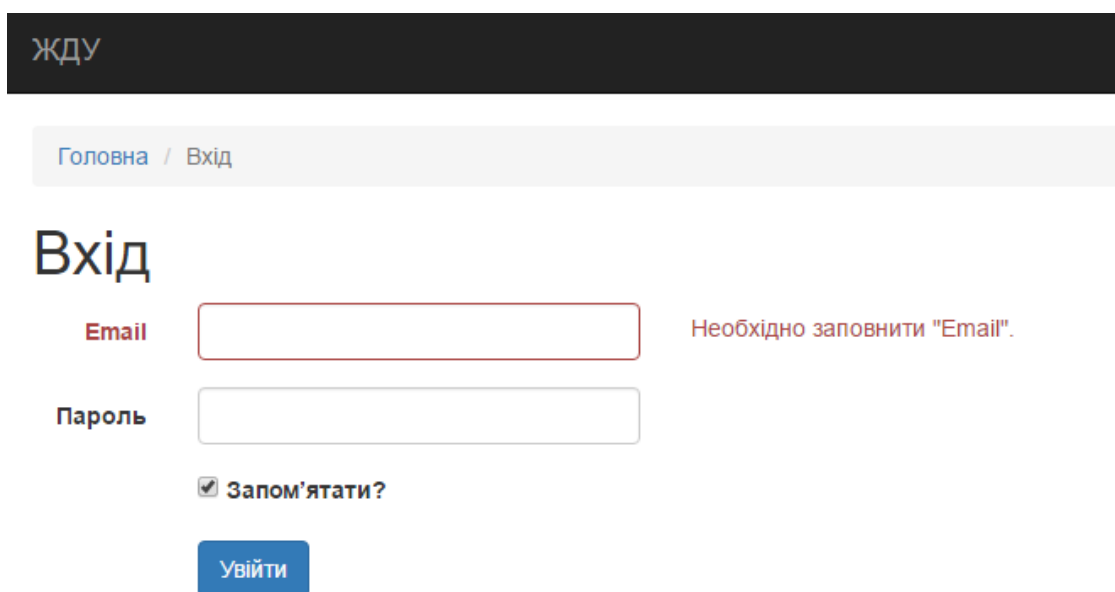
Фреймворк Yii2 містить функціонал авторизації в коробці. Тобто після встановлення ми маємо готовий функціонал авторизації. Отримати дані про поточного користувача ми можемо за допомогою компоненту `yii\web\User` через глобальний об'єкт `Yii`:

```
Yii::$app->user
```

Основні методи даного компоненту:

- `getId()` - повертає унікальний ключ для розпізнавання користувача;
- `getIdentity()`- повертає екземпляр класу поточного користувача з заповненими даними;
- `getIsGuest()` – перевіряє чи користувач авторизований.

Форма входу має наступний вигляд(поле Email має помилку валідації):



The image shows a login form with a dark header containing the text "ЖДУ". Below the header is a breadcrumb trail "Головна / Вхід". The main heading is "Вхід". There are two input fields: "Email" and "Пароль". The "Email" field has a red border and a red error message "Необхідно заповнити "Email"." to its right. Below the "Пароль" field is a checkbox labeled "Запом'ятати?". At the bottom is a blue button labeled "Увійти".

Після успішного входу користувач має додаткові можливості. Студенти можуть створювати заяви на вибір предметів на вивчення в наступному навчальному році(заява оформляється раз на рік, при повторній спробі оформлення заяви на вже заповнений рік буде видана помилка).

Адміністратор має додаткові функції для збору статистики заяв по рокам та налаштуванням системи(предмети, спеціальності та ін.).

Для контролю можливих дій користувачів використовується вбудований у фреймворк контроль доступу на основі ролей(Role Based Access Control, RBAC). Завдяки цьому ми можемо обмежити можливості студента, та видати помилки при переході на адміністративні розділи.

Усього система підтримує 3 види користувачів:

- гість – має доступ лише до інформації про сайт, сторінки входу та реєстрації;
- користувач(студент) – має доступ до генерації заяв(може створювати заяву раз на рік);
- адміністратор – бачить всі заяви, має доступ до редагування предметів, напрямів підготовки та генерування статистики.

### **3.4. Генерація заяв**

Основний функціонал системи полягає в генерації заяв студентів на наступний навчальний рік. Процес створення заяви складається з декількох етапів:

1. Реєстрація або вхід.
2. Налаштування профілю(факультет, курс, дані студентського квитка, форма навчання, напрям підготовки та спеціальність), дані профілю автоматично підтягуються в форму заяви.
3. Створення заяви. Для цього потрібно заповнити ті ж самі поля що й у профілі, а також обрати рік навчання(на один рік можлива лише одна заява).
4. Вибір предметів складається з 3 категорій(максимум система підтримує 5 предметів однієї категорії):
  - a. Дисципліни соціально-гуманітарної підготовки.
  - b. Дисципліни фундаментальної, природничо-наукової та загальноєкономічної підготовки.
  - c. Дисципліни професійної і практичної підготовки.

Після збереження користувач переходить в перегляд заяви, де може відредагувати, видалити або згенерувати файл заяви. Для генерації був обраний формат PDF.

PDF(Portable Document Format)[20] - це формат файлу, який використовується для подання документів незалежно від прикладного програмного забезпечення, апаратного забезпечення та операційних систем. Кожен файл PDF інкапсулює повний опис документу фіксованого макету, включаючи текст, шрифти, графіку та іншу інформацію, необхідну для його відображення.

Формат є відкритим, ,удь-хто може створювати додатки, які можуть читати і писати PDF файли без необхідності платити ліцензійні відрахування компанії Adobe Systems; Adobe має патенти на PDF, але ліцензії на їх



безплатне використання в розробці програмного забезпечення дотриманні у PDF-специфікації.

Для генерації PDF файлу за допомогою PHP була використана бібліотека mPDF[21]. Сама бібліотека представляє собою клас, котрий генерує PDF файл з HTML розмітки використовуючи кодування UTF-8.

В даний момент бібліотека потребує PHP 5.6+ з розширеннями mbstring та gd. Для встановлення потрібно виконати команду:

```
composer require mpdf/mpdf
```

Приклад згенерованої заяви(в сучасних браузерах необов'язково завантажувати файл для перегляду, він може відкриватись прямо у браузері):

Проректору з навчальної роботи  
Житомирського державного  
університету імені Івана Франка  
Янович Л.М.  
Коваленко Катерина Ярославівна  
студента 4 курсу  
історичного факультету  
денної форми навчання  
напряму підготовки 6.020302 Історія  
з подальшою спеціалізацією  
Правознавство

#### Заява

Відповідно до Положення про організацію освітнього процесу у Житомирському державному університеті імені Івана Франка прошу включити мене в групи для вивчення дисциплін за вибором студента на 4 курсі у 2016-2017 н.р.

**1.1. Дисципліни соціально-гуманітарної підготовки**

1. Соціологія

2. Релігієзнавство

30 серпня 2016 р.

Код генерації виглядає наступним чином:

```
<?php
/**
 * Generates PDF for an existing Claim model.
 *
 * @param int $id
 *
 * @return mixed
```

```

*/
public function actionExport($id)
{
    $model = $this->findModel($id);

    $content = $this->renderPartial('_export', [
        'model' => $model,
    ]);

    $PDFContent = mb_convert_encoding($content, 'UTF-8', 'UTF-8');

    $pdf = new Pdf([
        // set to use core fonts only
        'mode' => Pdf::MODE_UTF8,
        // A4 paper format
        'format' => Pdf::FORMAT_A4,
        // portrait orientation
        'orientation' => Pdf::ORIENT_PORTRAIT,
        // stream to browser inline
        'destination' => Pdf::DEST_BROWSER,
        // your html content input
        'content' => $PDFContent,
        // set mPDF properties on the fly
        'options' => ['title' => 'Заява №'.$model->id],
    ]);

    return $pdf->render();
}

```

Слід зазначити, що бібліотека хоч використовує HTML та CSS, але не має повної підтримки всіх можливостей розмітки(наприклад border-radius).

### 3.5. Сторінка налаштувань

Користувач адміністратор має додатковий доступ до сторінки «Налаштування». На цій сторінці знаходяться посилання для редагування актуальних даних системи, а також функціонал генерації звітів.

Повний список посилань виглядає наступним чином:

- Факультети/Інститути – список наявних факультетів/інститутів з можливостями створення, редагування та перегляду деталей;
- Кафедри – список наявних кафедр з можливостями створення, редагування та перегляду деталей;
- Дисципліни – список наявних дисциплін з можливостями створення, редагування та перегляду деталей;
- Напрями на спеціальності - список наявних напрямів підготовки з можливостями створення, редагування та перегляду деталей;
- Звіти – сторінка для генерування звітів по факультетам та спеціальностям.

















Всі сторінки(крім звітів) мають один вигляд(лише відрізняючись полями даних). Приклад вигляду сторінки факультетів на момент написання роботи:

Головна / Налаштування / Факультети/Інститути

## Факультети/Інститути

Додати

Показані 1-8 із 8 записів.

#	Назва	Тип	
	<input type="text"/>	<input type="text"/>	
1	Іноземної філології	Інститут	 
2	Історичний	Факультет	 
3	Педагогіки	Інститут	 
4	Природничий	Факультет	 
5	Соціально-психологічний	Факультет	 
6	Фізико-математичний	Факультет	 
7	Фізичного виховання і спорту	Факультет	 
8	Філології та журналістики	Інститут	 

### 3.6. Генерація звітів

Система має функціонал генерації звітів. Звіти діляться на 2 типи: по предметам та по студентам.

Форма генерації звіту має наступний вигляд:

Головна / Налаштування / Звіти

## Звіти

Тип звіту  
Звіт по предметам

Форма навчання  
денна

Курс  
3

Рік початку навчання на курсі  
2017

Факультет  
Оберіть факультет

Напря́м  
Оберіть напрям

Спеціальність/Спеціалізація

Згенерувати звіт

Звіт по предметам перевіряє список доступних для вибору та підраховує кількість студентів котрі обрали предмет. Звіт по студентам робить все те ж саме, але замість кількості студентів виводить список їхніх ПІБ.

Для генерації звітів та можливості подальшої обробки був обраний формат електронних таблиць Excel[22], а саме версія 2007 року. Для генерації була обрана бібліотека RHPExcel[23].

RHPExcel – це бібліотека, написана на чистому PHP, яка являє собою сукупність класів які дозволяють операції зчитування та запису для різних форматів електронних аркушів(spreadsheet): Excel (BIFF) .xls, Excel 2007 (OfficeOpenXML) .xlsx, CSV, Libre/OpenOffice Calc .ods та інших. Проект побудований навколо стандарту OpenXML Microsoft і PHP.

Повний список підтримуваних форматів на момент написання роботи:

Для читання:

- BIFF 5-8 (.xls) Excel 95 and above
- Office Open XML (.xlsx) Excel 2007 and above
- SpreadsheetML (.xml) Excel 2003
- Open Document Format/OASIS (.ods)
- Gnumeric
- HTML
- SYLK
- CSV

Для запису:

- BIFF 8 (.xls) Excel 95 and above
- Office Open XML (.xlsx) Excel 2007 and above
- HTML
- CSV
- PDF

Слід зазначити що для зміни формату звіту(мова йде про електронні таблиці) достатньо лише вказати новий формат при збереженні документа, функціонал генерації структури є універсальним для всіх підтримуваних форматів. Приклад коду збереження файлу, де ми обираємо вихідний формат:

```
<?php
// Save Excel 2007 file
header('Content-Type: application/vnd.ms-excel');
header('Content-Disposition: attachment;filename="'.FILE_NAME.'.xlsx"');
header('Cache-Control: max-age=0');
$objWriter = new \PHPExcel_Writer_Excel2007($objPHPExcel);
$objWriter->save('php://output');
```

Готовий звіт по предметам виглядає наступним чином(для звіту брались дані студентів 4 курсу історичного факультету, спеціальності 6.020302 Історія(з подальшою спеціалізацією Правознавство)):

	A	B	C	D	E	F
1	<b>Звіт</b>					
2						
3	<i>денна форма навчання</i>					
4						
5	<b>6.020302 Історія(з подальшою спеціалізацією Правознавство)</b>					
6	<b>№ п/п</b>	<b>Цикл/дисципліна</b>	<b>Курс/семестр</b>	<b>Кафедра</b>	<b>Кількість студентів</b>	
7	1	Дисципліни соціально-гуманітарної підготовки / Соціологія / Вибір №1	4 / 7	Філософії	12	
8	2	Дисципліни соціально-гуманітарної підготовки / Культурологія / Вибір №1	4 / 7	Естетики, етики та образотворчого мистецтва	35	
9	3	Дисципліни соціально-гуманітарної підготовки / Релігієзнавство / Вибір №2	4 / 7	Філософії	12	
10	4	Дисципліни соціально-гуманітарної підготовки / Геополітика і національні інтереси / Вибір №2	4 / 7	Філософії	35	
11						
12						

Як було сказано вище, звіт по студентам замість кількості виводить список ПІБ.

## ВИСНОВКИ

У даній роботі було проведено аналіз засобів розробки програмного забезпечення: викладено фрагменти історії розвитку засобів розробки, парадигми програмування та засоби реалізації. Були досліджені засоби розробки ПЗ за допомогою мови програмування PHP, використання фреймворку Yii2 та робота з IDE PhpStorm. В межах даної наукової роботи була розроблена автоматична система збору даних для студентів ЖДУ імені Франка, яка полегшує вибір предметів за бажанням студента, та оформлення необхідних заяв для вивчення цих предметів. Використовувалось інтегроване середовище розробки PhpStorm, мова програмування PHP 5.6, СУБД MySQL, фреймворк Yii2, система контролю версій Git та менеджер залежностей Composer.

Підсумовуючи, все зроблене, можна сказати, що робота над даним проектом допомогла мені краще зрозуміти механізм роботи веб-додатків, що є дуже важливим в такій сфері діяльності як програмування та автоматизація робочих процесів. Вивчення додаткової літератури не тільки збільшило мої знання, а й допомогло краще реалізувати практичне завдання.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. PHP: Hypertext Preprocessor [Електронний ресурс] – Режим доступу: URL : <http://php.net/> - Title from the screen.
2. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/PHP> – Title from the screen.
3. PHP: SPL – Manual [Електронний ресурс] – Режим доступу: URL : <http://ua2.php.net/manual/en/book.spl.php> - Title from the screen.
4. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/Yii> – Title from the screen.
5. Wikipedia - [Електронний ресурс] – Режим доступу: URL: [http://uk.wikipedia.org/wiki/Розробка\\_програмного\\_забезпечення](http://uk.wikipedia.org/wiki/Розробка_програмного_забезпечення) – Title from the screen.
6. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/Асемблер> – Title from the screen.
7. Н. Вирт. Систематическое программирование. – М.: Мир, 1977. – 183 с.
8. У. Дал, Э. Дейкстра, К. Хоор. Структурное программирование. – М.: Мир, 1973. – 247 с.
9. Э. Дейкстра. Дисциплина программирования. – М.: Мир, 1978. – 275 с.
10. Д. Ван Тассел. Стил, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1985. – 332 с.
11. М. Зелковиц, А. Шоу, Дж. Гэннон. Принципы разработки программного обеспечения. – М.: Мир, 1982. – 368 с.
12. Гутманс Э. PHP 5. Профессиональное программирование [пер. с англ.] / Э. Гутманс. – СПб.: Символ-Плюс, 2006. – 704 с.
13. Зандстра М. Р PHP. Объекты, шаблоны и методики программирования [пер. с англ.] / М. Зандстра. – СПб.: Вильямс, 2015. – 576 с.



14. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/Фортран> – Title from the screen.
15. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/PhpStorm> – Title from the screen.
16. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/MySQL> – Title from the screen.
17. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/phpMyAdmin> – Title from the screen.
18. PHP: Hypertext Preprocessor [Електронний ресурс] – Режим доступу: URL: <http://php.net/archive/2016.php#id2016-12-01-3> – Title from the screen.
19. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Посіб./ За ред. О.І. Пушкаря – К.: Видавничий центр "Академія", 2001. – 696 с.
20. Wikipedia - [Електронний ресурс] – Режим доступу: URL: <http://uk.wikipedia.org/wiki/PDF> – Title from the screen.
21. mPDF is a PHP library generating PDF files from UTF-8 encoded HTML - [Електронний ресурс] – Режим доступу: URL: <https://github.com/mpdf/mpdf> – Title from the screen.
22. Wikipedia - [Електронний ресурс] – Режим доступу: URL: [http://uk.wikipedia.org/wiki/Microsoft\\_Excel](http://uk.wikipedia.org/wiki/Microsoft_Excel) – Title from the screen.
23. PHPOffice/PHPExcel: A pure PHP library for reading and writing spreadsheet files - [Електронний ресурс] – Режим доступу: URL: <https://github.com/PHPOffice/PHPExcel> – Title from the screen.