**ADVANCE PROGRAMMING**

MUHAMMAD NOUMAN | BSCS- 7B | REG # 218663

**ASSIGNEMENT #1**

**INTRODUCTION**

Yii is a dynamic and efficient php framework that is used to make excellent web applications based on a basic template. It is fairly scalable and secure and its database is very convenient because it reads data in form of objects. It helps the programmer as he doesn’t have to write repetitive sql statements.

**INSTALLATION PROGRESS**

Firstly I downloaded *“Composer.exe”* which is package manager for yii. It is not only use to download all of the yii template, folders but also used for downloading and integrating plugins/ extensions.

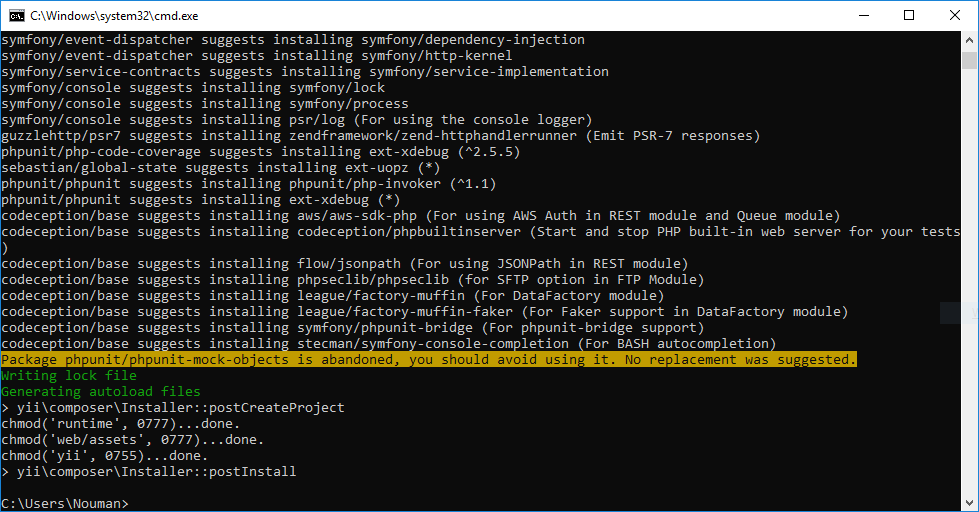
In order to install yii via composer I typed this statement on the terminal:

composer create-project --prefer-dist yiisoft/yii2-app-basic basic

This once executed, creates the directories that would be needed to display the basic template and modify accordingly. The three most important directories are model, view and controller. This is because yii is based on MVC architecture.

In order to configure php server I ran this command:

php yii serve



**EXPLORING IT WITH USER PERSPECTIVE**

Then I entered this URL:

<http://localhost:8080/>

In order to view the basic template that is provided by yii. I explored this template by modifying its associative files. For Instance, I was able to modify the “About” page by going to “about.php” file and changing the code so it would display my name on the web page.

The other work that I done was to explore all files and watched how they are interlinked and packed in zip file i.e. documented as well.

**DATABASE AND CODING STRUCTURE**

For learning the coding structure of Database I installed *PHP Data Objects* and driver for MYSQL which is “pdo\_mysql” .

After installation of that I went to *config/db.php* and typed the following code:

**<?php**

**return** [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=yii2basic',

'username' => 'root',

'password' => '',

'charset' => 'utf8',

];

By using the class based approach I added new records because yii deals with data in form of objects which is super good benefit of it. It prevents writing same repetitive sql queries. I also learned how to fetch data and display pages with the help of [yii\data\Pagination](https://www.yiiframework.com/doc/api/2.0/yii-data-pagination) and [yii\widgets\LinkPager](https://www.yiiframework.com/doc/api/2.0/yii-widgets-linkpager) and building custom views and functions.

**ONLINE PLUG-INS AND MODULES**

**JSON EDITOR**

I used JsonEditor widget for Yii2 Framework uses Json Editor plugin for jQuery. A complete description about the extension is described below:

**INSTALLATION**

It was installed using the same composer software. For this the below code was run:

composer require slavkovrn/yii2-jsoneditor

**USAGE**

For using it I used the following code:

**<?php**

**use** **slavkovrn**\**jsoneditor**\**JsonEditorWidget**;

**use** **yii**\**widgets**\**ActiveForm**;

**use** **yii**\**helpers**\**Html**;

**?>**

**<?php** $ form = ActiveForm::begin(); **?>**

**<?**= $form->field($model, 'username')->widget(JsonEditorWidget::class,[

'rootNodeName' => 'root1',

]) **?>**

**<?**= $form->field($model, 'email')->widget(JsonEditorWidget::class,[

'rootNodeName' => 'root2',

]) **?>**

<div **class**="**form**-**group**">

<?= **Html**::**submitButton**('**Save**', ['**class**' => '**btn** **btn**-**success**']) ?>

</**div**>

<?**php** **ActiveForm**::**end**(); ?>​

**CONCLUSION**

Yii is an MVC based framework. Issue that I faced was many documentations of the extensions were in Chinese so I have to leave them. Else the framework was good to learn and provided with extensions that are handy for practical use like editors for custom work and chat extensions for providing with interconnection responsibilities.