|  |
| --- |
| Developer Documentation |
| Hela Weda Piyasa Ayurwedic Hospital |
| **Team Software Project – Group 08**  **Level – 3**  **Department of Computer Science**  **Faculty of Science**  **University of Jaffna** |

**Executive Summary**

* **Outcome to be achieved**

At present, there are lots of private hospitals are available to provide medical services. People are seeking private hospitals as they believe that the private hospitals are providing higher quality treatment than government hospitals. This is the online system that can bring more advantages for the doctors, patients etc.

* **Users**

There are mainly 5 users. They are Patient, Doctors, Pharmacist, Medicine Producers, Ingredient suppliers. Additionally, there is an admin also available to control some particular activities.

* **Problem Statement**
* **Patients**
* Nowadays, if a patient wants to see a doctor, they should book before. So they need to visit to hospital and book.
* If they want to buy medicines they need to go to that pharmacy.
* There is no way to send their current condition to corresponding doctors.
* Now, we have clinic books to see past histories. In case, if the book is missed, no way to get back.
* Doctors
* Normally, doctors write prescription in papers. So patient should carry the paper to pharmacist.
* To keep admitted patient record we need to maintain a big ledger book.
* Doctors want to update the admitted patient records to give medicines to them.
* If doctor will be late or unable to come, he should inform to the staffs.
* Doctor doesn’t aware himself how many patients are going to come today.
* Pharmacist
* Pharmacist is not aware of medicine stocks. When will be expired? , how many stocks are there?
* Pharmacist should ask with patient the prescription or medicine details and he need to calculate the bill.
* Pharmacists are unaware about the amount of medicines and expiry date of medicines available in the stock.
* Medicine Producer
* He should maintain a medicine table manually.
* He should maintain ingredient table manually.
* If he creates a new medicine he should acknowledge to pharmacists.
* When ingredients finish in stock he will say to suppliers.
* He may be forgotten to issue order. So he should maintain orders table manually also which is from pharmacist.
* Ingredient Suppliers
* He should manage what ingredients stock has and maintain orders from producers.
* **Solution**
* Patients
* Patients can simply book a doctor and order medicines from anywhere rather than go directly to the hospital.
* They can send their symptoms, as text, images even as audio file.
* No need to check any clinic books to see what the doctor said in last meeting.
* Doctors
* Doctors can pass the prescription to pharmacist via online.
* Patients can be admitted at hospital via/through this system.
* Daily updates also can be added through system.
* Doctors can see patients’ symptoms through this system and they can suggest the solution.
* Simply they can watch how many appointments they have today.
* They can update the availability time.
* Pharmacists
* Pharmacists are going to aware of medical stock through this system.
* They don’t need to calculate the bill amount. System will do.
* They can see the requested orders from patients and doctors.
* They can order medicines via online to medicine producer.
* Medicine Producer
  + Medicine and ingredient table will be maintained in their database.
  + He can able to acknowledge to pharmacist, when pharmacist requires new medicine.
  + Orders from pharmacist will be maintained.
  + Producer can place order to supplier when ingredients are finished.
* Ingredient Supplier
* Supplier can manage his ingredient stocks and can see the ingredients order details from medicine producer.

**Team Information**

This web system is developed by the following team members.

1. Elackshana M. - [elackshana@gmail.com](mailto:elackshana@gmail.com) 2017CSC018(Database Design)
2. P.A.Gamage. - [ajanthasirigamage@gmail.com](mailto:ajanthasirigamage@gmail.com) 2017CSC033(App Development)
3. Dilki Hasara - [dhwways@gmail.com](mailto:dhwways@gmail.com) 2017CSC022(UI Design)
4. S. lokavannilavan - [lokavan007@gmail.com](mailto:lokavan007@gmail.com) 2017CSC047(System Testing)
5. ULM Afrid - [mafrid029@gmail.com](mailto:mafrid029@gmail.com) 2017CSC029 (Team Leader)

Github Repository link : <https://github.com/afrid029/TeamSoftwareDevelopmentGroup08.git>

Mentor: Ms. Samantha Tharani Jayakumar.

**Software License**

We are using following softwares.

* PHP – version 7.4.10
* Laravel – version 4.1.1
* Composer – version 1.10.13
* Apache – version 2.0
* Microsoft Visual Studio Code – version 1.52.1
* Mysql – version 10.4.14

**Installing Softwares**

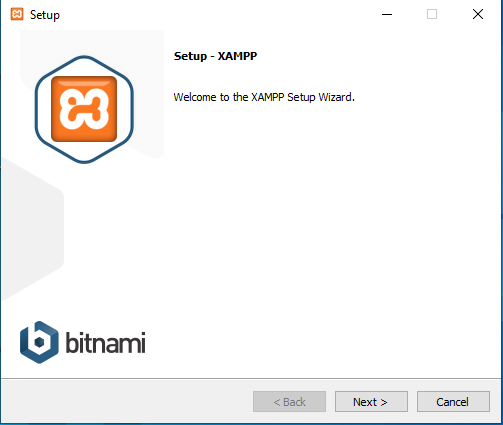
* **Installing PHP, Apache, Mysql**

Above three softwares will be installed by install **XAMPP**

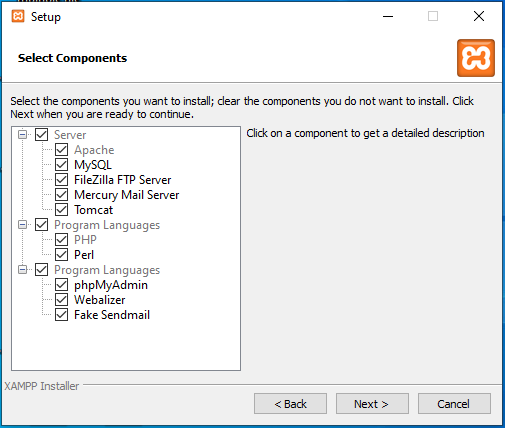
Download XAMPP here: <https://www.apachefriends.org/index.html>

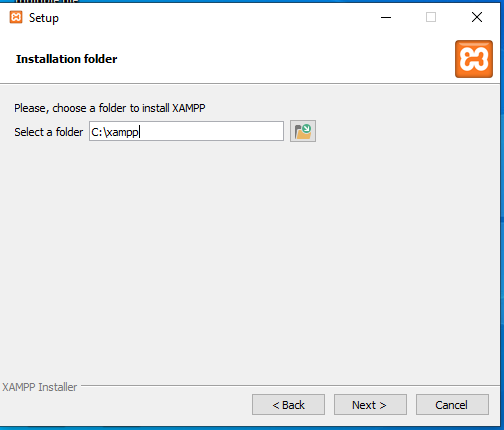
**Install XAMPP on windows**

Press **Next>**

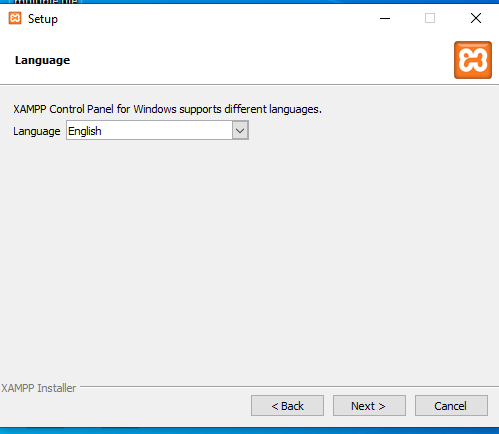


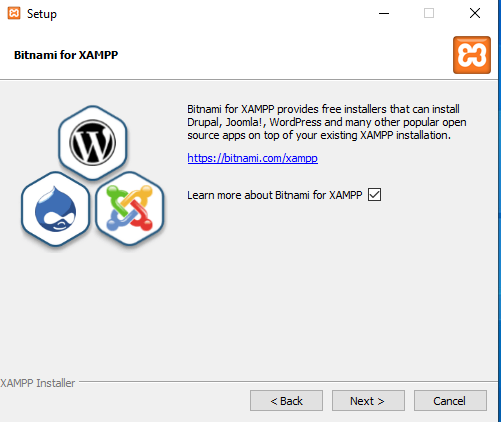
Press **Next>**

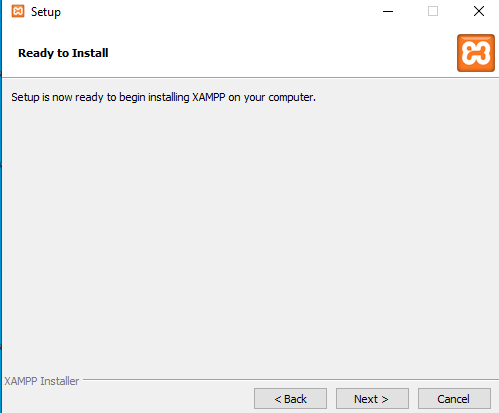


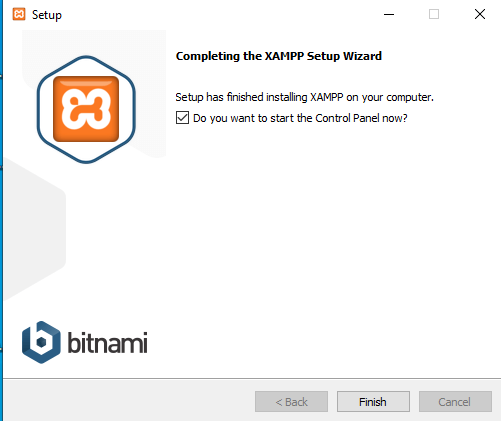
 Select a folder to install and press **Next>**

Select Language and press **Next>**

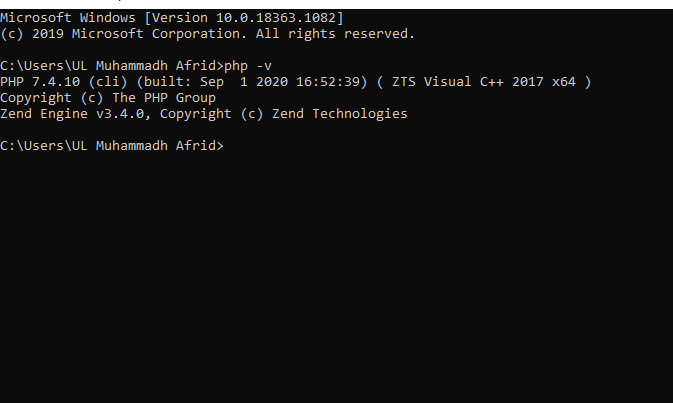










Check whether it is installed or not

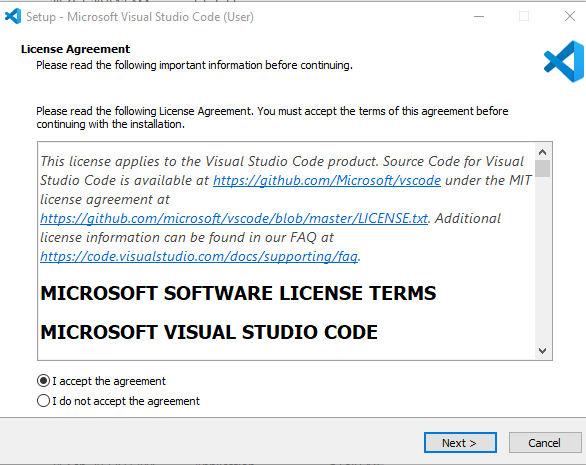
Open **cmd**  and type **php –v** to check the version of php

* **Installing Microsoft VS Code for windows**

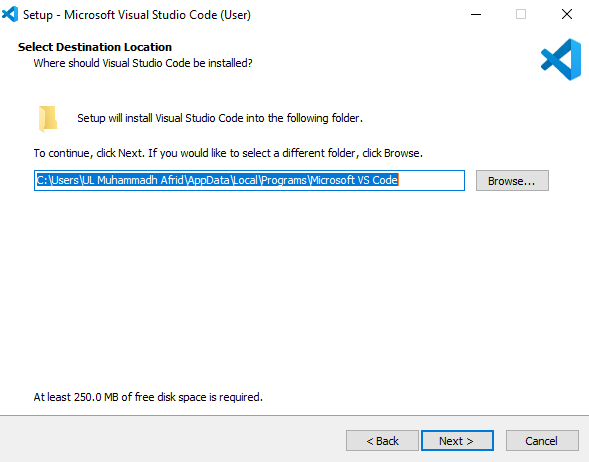
We use MS VS code as our code editor.

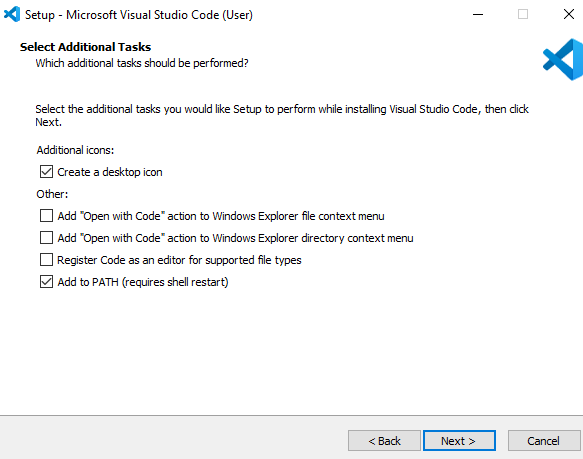
Download the software in : <https://code.visualstudio.com/\>

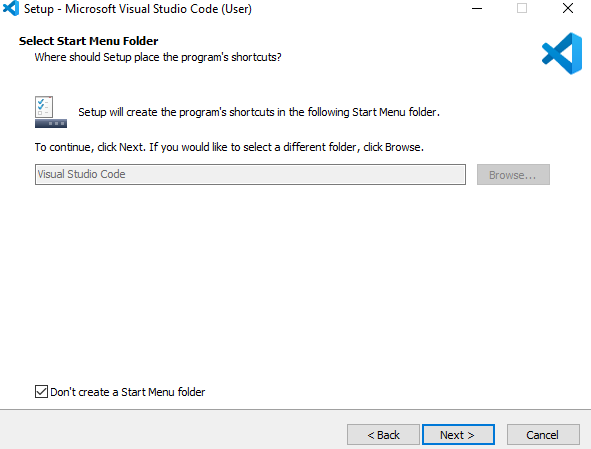
* **Agree** the agreement and start to install

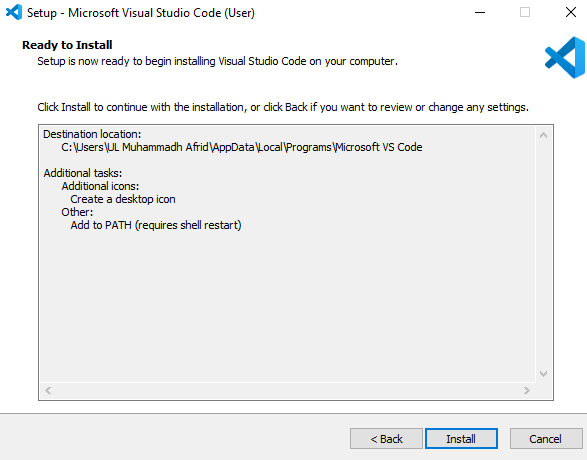


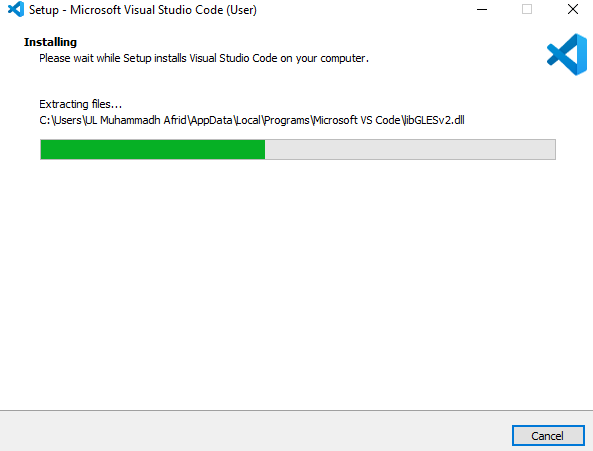
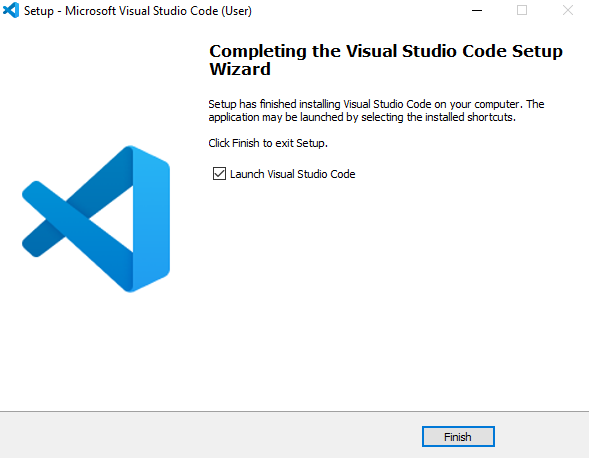
* Choose the **path** to install







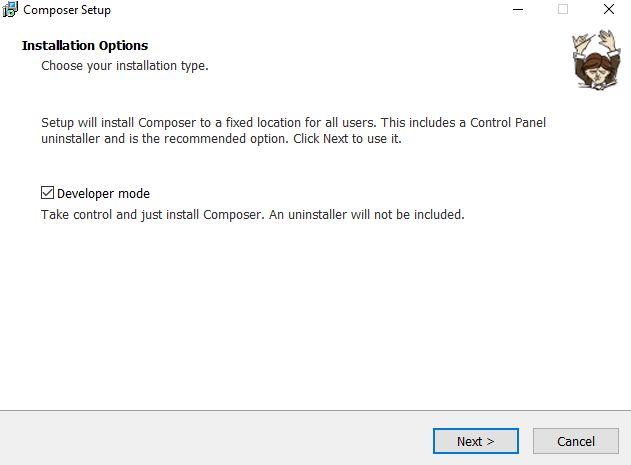


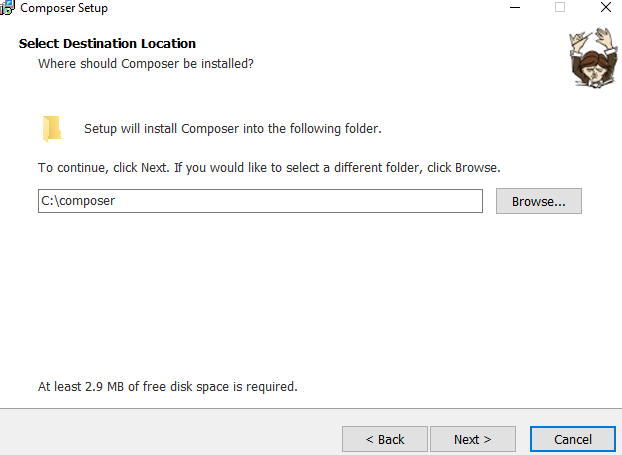


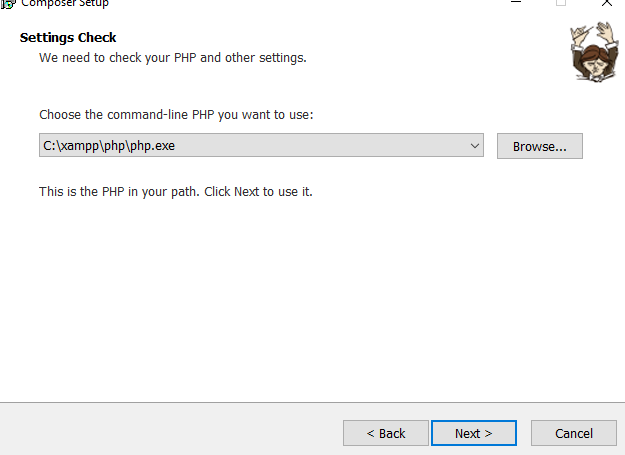
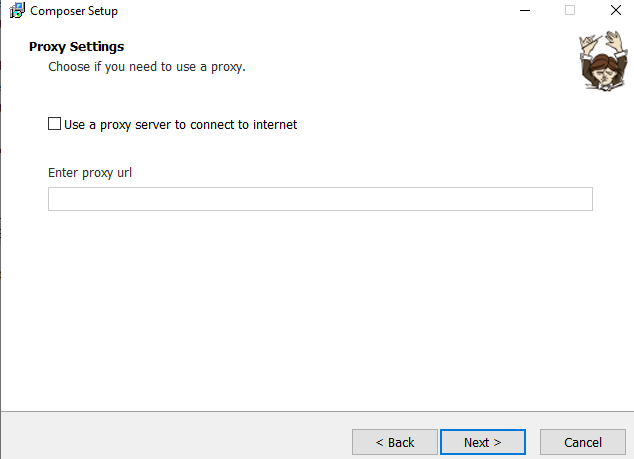
* **Installing Composer**

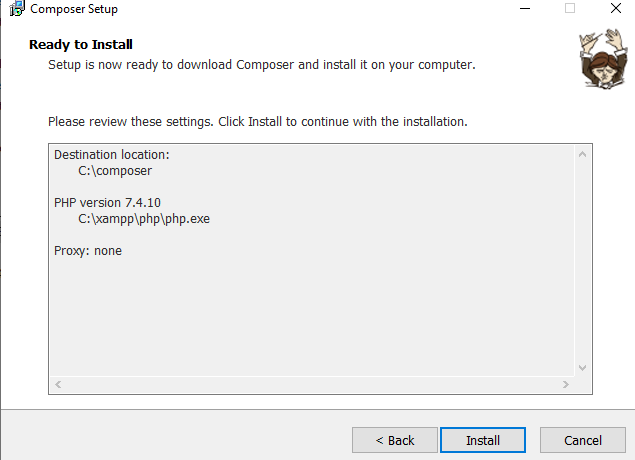
Composer is a dependency manager which contains libraries.

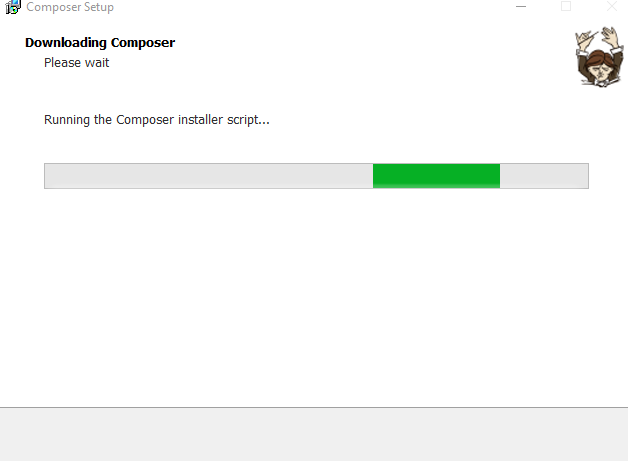
It is used to install packages like laravel framework, Doctrine, lodash etc..

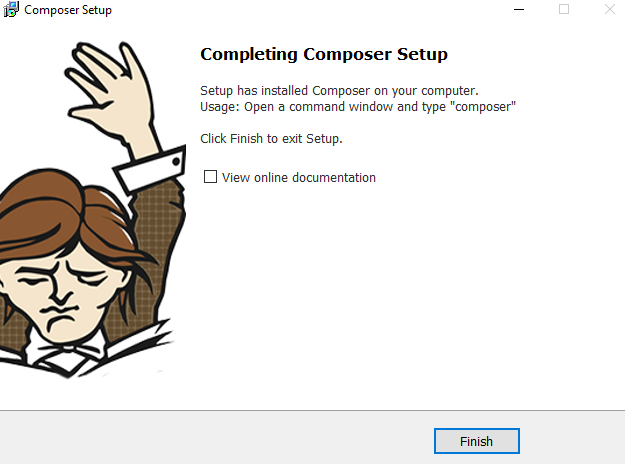
****

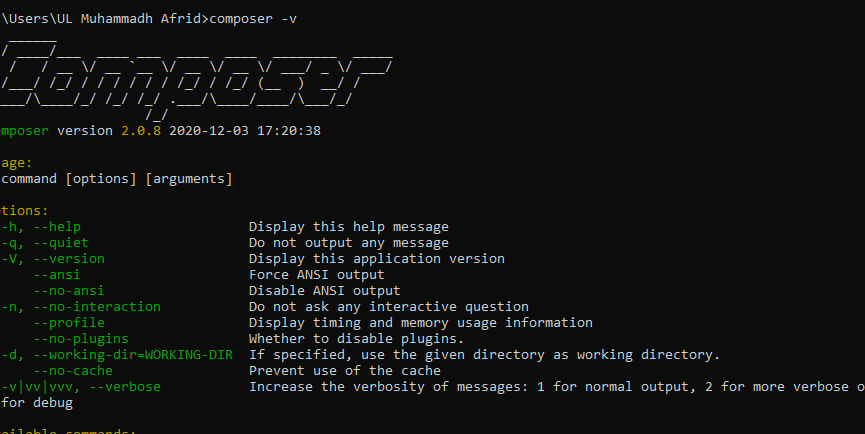
****

****



* 



****To check whether composer is installed or not : open **cmd** type **composer –v**

* **Installing laravel**

Laravel is installed using composer in cmd or visual code terminal.



A new project created by, (Laravel should be installed already)

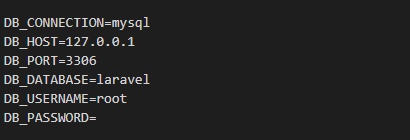


Or we can simply create project using composer command,

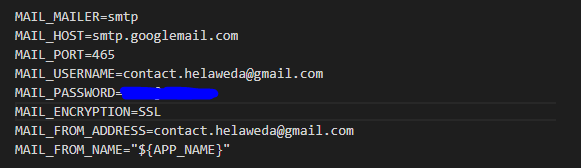


**Set up .env**

Set environment for database with database name.



Set environment to send email



**Basic Codes**

* **php artisan serve** - Run the project in localhost at 127.0.0.1 and port number 8000
* **php artisan make:controller <controller name>** – Create controllers.
* **Php artisan make:model <model name>** - Create models.
* **Php artisan migrate** – Migrate database model with tables.
* **Php artisan migrate:fresh** – Drop all database models and remigrate.
* **Php artisan db:seed** – Record data into database from seeder

**How to clone the project from my github repository.**

When you create a repository on GitHub, it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.

This procedure assumes you have already created a repository on GitHub, or have an existing repository owned by someone else you'd like to contribute to.

1. On GitHub, navigate to the main page of the repository.

2. Under the repository name, click Clone or download.

3. In the Clone with HTTPs section, click to copy the clone URL for the repository.

4. Open Git Bash.

5. Change the current working directory to the location where you want the cloned directory to be made.

6. Type git clone, and then paste the URL you copied in Step 2.

7. Press Enter. Your local clone will be created.

**URLs and methods.**

* /welcome – get method – Home page.
* /register – get method – Register as a patient.
* /saveuser – post method – submitting registering details of patients.
* /login – get method – login page.
* /log – post method – submitting login form details.
* /enter/id – get method – when open a browser in case if user did not logout from last login, it will redirect to corresponding user profile.
* /logout – get method – logout function.
* /forgotp – get method – redirect to forgot password action doing page.
* /forgotpass – post method – submit details from forgot password page.

**Doctors**

* /docedit – post method – when a doctor edits his profile data go through this link.
* dochome/{c}/ – get method – redirect a doctor to his home page.
* prescription/{c}/ – get method – redirect a doctor to prescription page.
* admitted/{c}/ – get method – redirect a doctor to admitted patients’ records page.
* available/{c}/– get method – redirect a doctor to available time periods page.
* addpatdetails/{c}/ – get method – redirect a doctor to patient admitting page.
* /savepres – post method – saves a prescription that doctor wrote.
* /saveadmit – post method saves a record about admitted patient.
* /pressearch – post method –returns search results of prescriptions.
* /adsearch – post method – returns search results of admitted patients’ records.
* /saveavailable – post method – saves an available time period.
* /avedit – post method – to edit an available time period.
* /avdelete/{id}/{docid} – get method – to delete an available time period.
* /patadmit– post method –saves details of newly admitted patient.
* /admitsearch – post method – returns search results of admitted patients.
* docsymp/{c}/ – get method – redirects a doctor to medical symptoms page.
* docviewSymp/{i}/{j}/ – get method – redirects a doctor to view details of a medical symptom.
* /docreply – post method – update doctor’s reply for a medical sympytom.
* appointment/{c}/– get method – redirects a doctor to appointments page.
* /appsearch – post method – returns search results of appointments.
* /docpic – post method – update profile picture of a doctor.
* discharge/{c}/{d} – post method –discharges an admitted patient.

**Patients**

* /pathome/pat1 – get method – a patient after login, redirect to his profile.
* /patedit – patch method – when a patient edits his profile data go through this link.
* /changeprofile/pat1 – patch method – patient change his profile picture.
* /symp/pat1 – get method – patient (pat1) is redirected to symptom adding page.
* /addsymptomps/pat1 – post method - Patient (pat1) is submitting his medical symptoms to doctor.
* /order/pat1 – get method - patient (pat1) is redirected to order medicine page.
* /ordermedicine/pat1 – post method – order medicine to pharmacist.
* /book/pat1 – get method – redirect a particular patient into book a doctor page.
* /showAvail – get method – filter doctors’ available time by patient wish
* /appoint – get method – redirect to page which shows available time for doctors.
* /confirmAppoinment – post method - when a patient confirm his appointment, it will be sent through this link.
* /deleteAppoint – get method – delete doctors’ appointment.
* /history/pat1 – get method – a patient see his medical history.

**Pharmacist**

* /phahome/pha1 – get method - after login as pharmacist, redirect to pharmacist profile page.
* /phaedit- post method - when a pharmacist edits his profile data go through this link.
* /pharprofilepic/pha1 – patch method – pharmacist change his profile picture.
* /medicalstock/pha1 – get method – redirect a pharmacist to medical stock page.
* /addmedicine – post method – Add new medicine into pharmacist medical stock.
* /updatemedicine/pha1 – post method – update existing medicine stock details in pharmacist medicine stock.
* /Delmedicine/pha1 – post method – Delete a medicine from pharmacist medical stock.
* /issueMedicine/pha1 – get method – redirect a pharmacist to issue medicine page.
* /Issuepatorder/pha1 – post method – issuing medicine order from patient online orderings.
* /issuedocorder/pha1 – post method – issuing medicine from doctor’s prescription.
* /phaordermedicine/pha1 – get method – redirect pharmacist to pharmacist medicine ordering page.
* Ordertopro/pha1 – post method – order medicine to producer from pharmacist.

**Medicine producer**

* mphome/{c}/ – get method – redirects a medicine producer to his profile home page.
* issuemedicine/{c}/ – get method – redirects a medicine producer to issue medicine orders page.
* Ingstock/{c}/ – get method – redirects a medicine producer to ingredients stock page.
* medstock/{c}/ – get method – redirects a medicine producer to his medicine stock page.
* ordering/{c}/ – get method – redirects a medicine producer to order ingredients page.
* /proedit – post method – edits profile of medicine producer.
* /proaddmedicine – post method – add new medicine stock.
* /proupdatemedicine – post method – update existing medicine stock details.
* /promeddelete/{c}/{d} – get method – delete a medicine stock.
* /proadding – post method – add a new ingredient stock.
* /proingdelete/{c}/{d} – get method – delete an ingredient stock.
* /proupdateing – post method – update details of existing ingredient stock.
* /proingorder – post method – place an ingredient order to ingredient supplier.
* /ordersearch – post method – return search result of ingredient orders.
* /issusearch – post method – return search result of issue medicine.
* reorder/{c}/{d} – get method – issue a medicine order.
* medicines/{c}/ – get method – redirect a medicine producer to medicine details page.
* /newmedicine – post method – add a new medicine.

**Ingredient supplier**

* suphome/{c}/ – get method – redirect an ingredient supplier to his profile home page.
* /supedit – post method – edit profile of ingredient supplier.
* /suppic – post method – edit profile picture of ingredient supplier.
* issueing/{c}/ – get method – redirect an ingredient supplier to issue ingredient orders page.
* newing/{c}/– get method – redirect ingredient supplier to add new ingredient page.
* supreorder/{c}/{d} – get method – issue an ingredient order.
* /newingredient– post method – add new ingredient.
* /issuingsearch – post method – return search results of ingredient orders.

**Admin**

* /adminpage/adminid – get method – redirect to home page of admin.
* /admedit – post method – edit profile of admin
* /regist/adminid – get method – redirect to users’ registration page.
* /addnew – post method – create new user (doctor/pharmacist/medicine producer, ingredient supplier).
* /profit/adminid – get method – redirect to profit view page.
* /patbill – post method – when admin use any filters in viewing profits from patient medicine orders.
* /patbill – get method – when admin search with no filters in patient medicine orders, it will provide all details.
* /docbill – get method - when admin use any filters in viewing profits from doctors medicine prescription.
* /docbill – get method – when admin search with no filters in doctors medicine prescription, it will provide all details.
* /profview/adminid – get method – see users’ profile by Admin.

**Database models**

* AllUsers : Allusers details.
* Doctor: save doctors’ details
* Ingredient\_supplier: save supplier details.
* Medicine\_producer: save medicine producer’s details.
* Patient: save patients’ details.
* Pharmacist: save pharmacists’ details.
* admin: save admin details.
* Add\_pat\_up : Admitted patients updates.
* Add\_pat : Add new patient to ward.
* add\_symptomps : Store medical symptoms
* Doc\_available\_time : save doctors’ available time
* Ingredient\_ordering : save ingredients orders details from medicine producer.
* Ingredient\_stock : ingredient stock details in producer.
* Ingredients: maintain all ingredient details by supplier.
* Medical\_history : medical histories of outpatients.
* Medicine\_ordering : save medicine orders details from pharmacist to producer.
* Medicine\_stock : medicine stocks pharmacist.
* Medicines: maintain all medicines details by medicine producer.
* new\_med\_stock : medicine stock of medicine producer.
* OnlineBooking : save patients’ online booking details.
* pat\_med\_ordering : save patients’ medicine ordering details.

**Controllers**

* AdminController : control all admin backend activities.
* patientsController : control all patients backend activities.
* pharmacistController : control all pharmacists backend activities.
* login : control login parts’ backend activities of all users.
* PageController : control redirecting activities of login, register and logout functions.
* redirect : control all backend activities of redirecting parts pages of doctors, medicine producer and ingredient supplier.
* search : control backend activities of searching functions of doctors, medicine producer and ingredient supplier.
* store : control backend activities of new record creating functions of doctors, medicine producer and ingredient supplier.
* update : control backend activities of update existing records functions of doctors, medicine producer and ingredient supplier.