IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: J.A. Mujeeb
Student Reference Number: 10707284

Module Code: CNET343SL	Module Name: Distributed Systems
Coursework Title: Weather Reporting System Proposal	
Deadline Date: 03/12/2021	Member of staff responsible for coursework: Mr. Pramudya Thilakaratne.
Programme: BSc (Hons) Plymouth Software Engineering	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

J.A. Mujeeb – 10707284
G.M.D.D. Ratnayake – 10707351
S.O. Perera – 10707315
N. S. De Alwis – 10707160
M. D. A. Medhavi – 10707278
P. P. L. Dilhani – 10709402

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: J.A. Mujeeb

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed: 

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

- *Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

Contents

Acknowledgement

First and foremost, we'd like to extend our sincere gratitude towards Mr. Pramudya Thilakaratne, our module lecturer. We are extremely humbled and grateful to have been able to receive his mentorship, guidance, and support.
The overall accomplishment of this project demanded a significant amount of guidance from many individuals. As a team, we are extremely fortunate to have had this from start to finish.
Finally, we would not have been able to successfully complete this assignment without the challenging work and assistance of all the team colleagues itself. We all enjoyed collaborating with each other.

Introduction to the Distributed System

The use of weather reporting in day-to-day life is especially important. Its utilization could greatly influence the outcome of a scenario. It could be something as simple as deciding whether you should take your umbrella on your way out, or even as major as handling cultural operations, agriculture and farming or livestock protection implementations.
With Sri Lanka's weather rapidly varying, weather can transition from largely homogeneous temperatures to torrential rain in a heartbeat. With the adaptation of unexpectedly frequent changes, it is crucial to make sure to be prepared.
Sri Lanka is liable to prevailing and predictable effects of climate change. Preceding natural disasters (such as tsunamis, floods, landslides, droughts, and cyclones) can greatly substantiate this. Profoundly weather-sensitive sectors in Sri Lanka include transport, agriculture, construction, energy, and disaster risk management.
It is important to note that even with weather stations, live broadcasts, and television/radio stations existent, not all are aware of the continuously differentiating weather conditions.
The impacts and effects of climate change could invite climate change-induced hazards and disasters. The unawareness of weather reports may affect the lives of many. Thousands of citizens are prone to being affected by the threat of climate change. They may find themselves in life threatening situations. Weather prediction is essential in order to provide citizens with pragmatic information. Furthermore, this also aids in the reduction of weather-related losses, personal safety and health, enhancement of societal benefits and in supporting economic prosperity.
It is abundantly clear that weather forecasting reports are essential to mitigate the effects civilians getting caught in a severe crisis.

Introduction to Quick Weather –

What is distributed system?

A distributed system is a collection of independent computers that appears to its users as a single coherent system. (Tanenbaum)
We have a system or application that could be a software. In that particular system we have a number of independent computers (individual computers inside the system). These individual computers are doing their own task in the system, but for the end user this looks like one single system. Those computers are doing their own tasks but for the user, they can see everything as one system.
A distributed system is contrived to assist the development process of services and applications. These services and applications are able to manoeuvre a physical architecture that would contain multiple independent processing elements. The

processing elements do not share primary memory but do partake in complying with dispatching asynchronous messages via communication network.
Furthermore, Distributed Systems also utilize independent hardware in furtherance of creating a software. It would appear as a complete system all working as a single unit, but it makes use of multiple computers in contemplation of making a distributed system function.
The primary memory of other hardware elements is not employed by Distributed Systems. Instead, multiple independent processing elements within the hardware are utilized. Using asynchronous messages, they can communicate over a networked communication.

1.1 What is Quick Weather?

In the following report, we would like to highlight the main outcome of our system. To make the reader of this report get a clear idea, we have simply built a Web Application and Mobile Application that will be running as a client application.
The public (users) often need to be informed on the current weather situation. Yet, as it is reckoned that Sri Lanka is still a developing country, it is only specified in either the news or radio stations.
In normal circumstances, the user may only be able to view the weather around their current location, but with Quick Weather, any user can view the weather around the country.
Our goal is to present a high-performance Distributed Weather System that encompasses paramount customer satisfaction, convenience, imperative security, user-friendliness, and maximum authenticity.
This system will present the needed weather for any planned activities in their personal lives. Furthermore, registered users have the ability to publish and access Blog Posts within the website. This feature will aid in giving any user a more visual idea on what the weather situation is like in that particular area. Blog Posts also anchor in many benefits, as it is a superb way to drive traffic to the website and also in building credibility.

Usefulness of the application to the real-world

Quick weather can perform 03 main functions

- Users possess the ability to yield a weather forecast at any location within the country with the utmost accuracy and efficiency of the mobile and web application. The user simply must enter the name of the location in it to access weather details**.**
- Registered users can upload and view posts about weather from the mobile and web application. This feature will aid in giving any user a more visual idea on what the weather situation is like in that area.
- Users can view post descriptions by using the standalone application with NetBeans.

Scope of Project

The following is a list of the application's scope:

- This program can show the state of the atmosphere for a specific place.

- By checking uploaded posts (uploaded by registered users), it is possible to take precautionary steps against catastrophic rains, winds, extreme high or low temperatures, diseases, and pests.
- Distribution Scalability.
- A high-performance system that encompasses paramount customer satisfaction, convenience, imperative security, user-friendliness, and maximum authenticity.
- A Desktop Application, Web Application and Mobile Application.

1.2 Diagrams

1.2.1 System Diagram

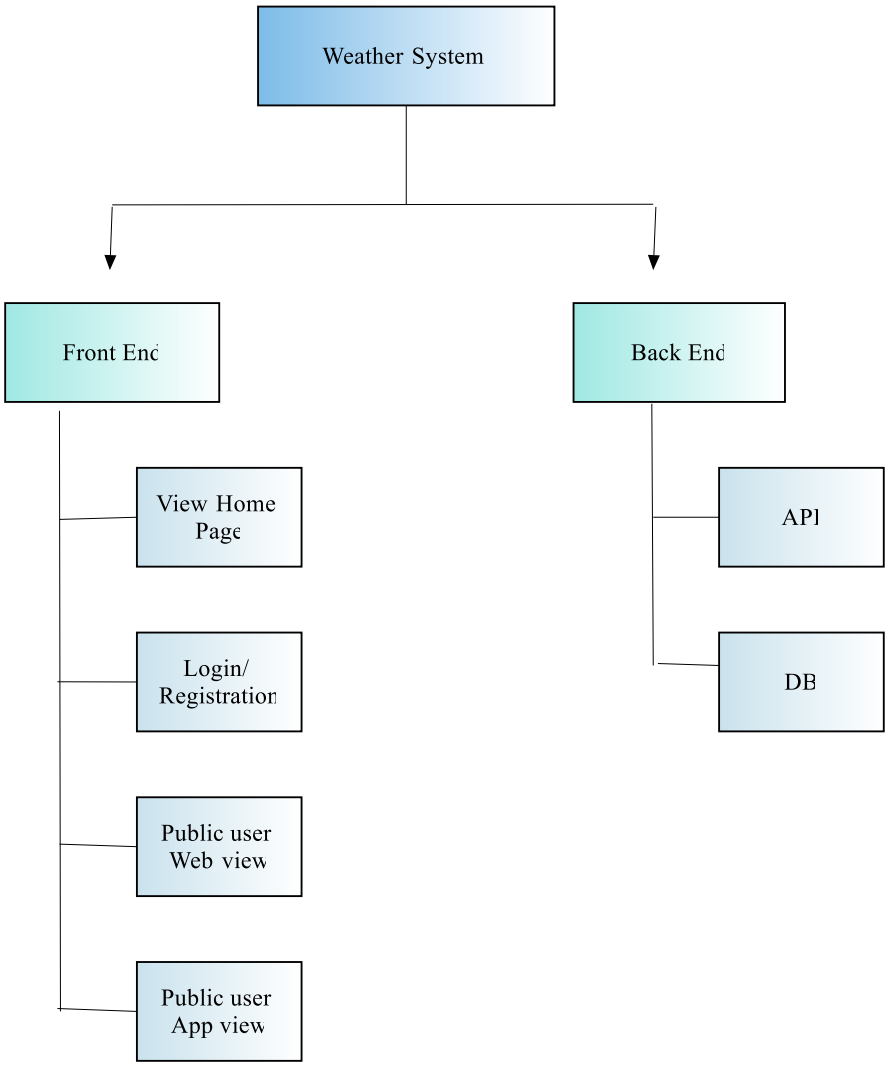


Figure 1: System Diagram

1.2.2 Context Diagram

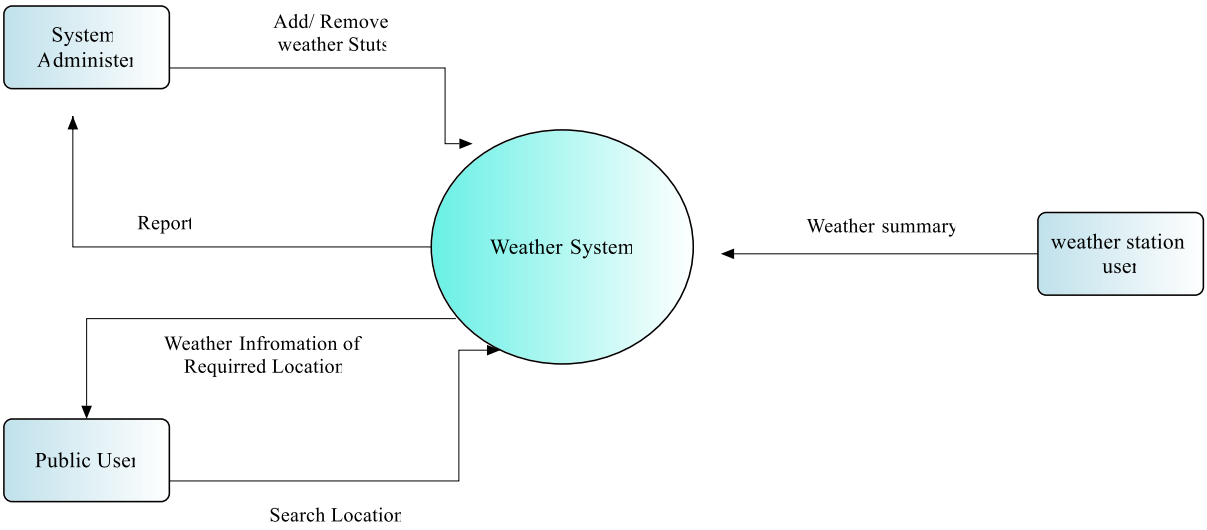


Figure 2: Context Diagram

1.2.3 Use Case Diagram

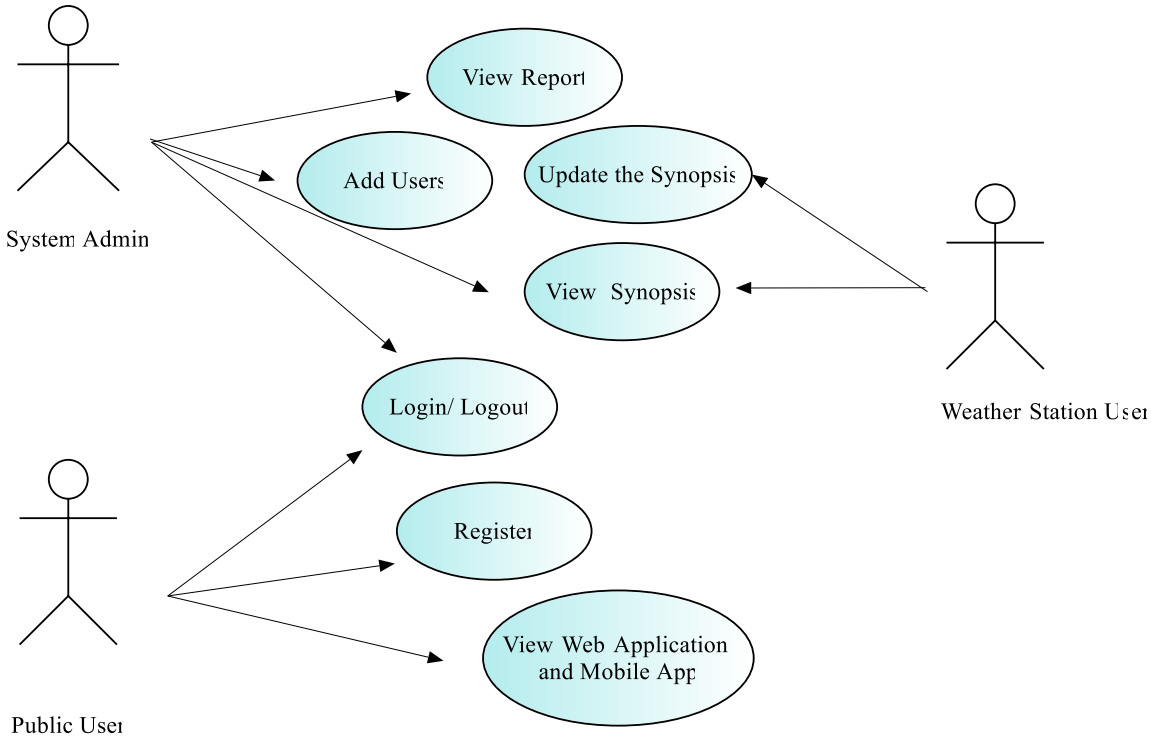


Figure 3: Use case Diagram

Mobile Application as Client

The Quick Weather mobile application is created using Android Studio, the language that Android Studio uses is Java. Since Android Studio has SDKs that run emulators, it is quick and easy to use.
The client can login/signup and post any type of image related to Technologies used:

- ✓ Laravel.
- ✓ Java in web and mobile application.
- ✓ MySQL for the database.

Methodology

Keyword	Category	Description
Java NetBeans	IDE	Utilized for the standalone Desktop Application.
PHP Laravel	Framework	Used for the API construction and Web Application Development.
PHP Laravel	Load Balancer	Distributes the web traffic amongst two or more servers and are often used for websites which receive high volumes of traffic.
Android Studio	IDE	Android studio has in built emulators which makes the system faster and more advantageous to developers.

Table 1: Methodology

Database: PhpMyAdmin MySQL

As PHP is utilized to code the Web Application, it was more coherent to utilize PhpMyAdmin for MySQL database management. PhpMyAdmin is an open-source third-party software tool, written in PHP. The primary objective of utilizing phpMyAdmin is to superintend and operate the administration of MySQL over the web. It is also feasible to run CRUD operations like databases, copy, tables, rename, databases, tables, columns, etc. phpMyAdmin can run on any server or any OS as it has a web browser. With the utilization of phpMyAdmin, it is possible to edit, create or delete the database without much difficulty. In collation to the MySQL command-line editor, it is easier to manage elements with the utilization of the phpMyAdmin graphical interface. Several servers can also be operated simultaneously. Data can also be exported into various formats like Word, PDF, SQL, XML, Spreadsheet, etc.

Gantt Chart

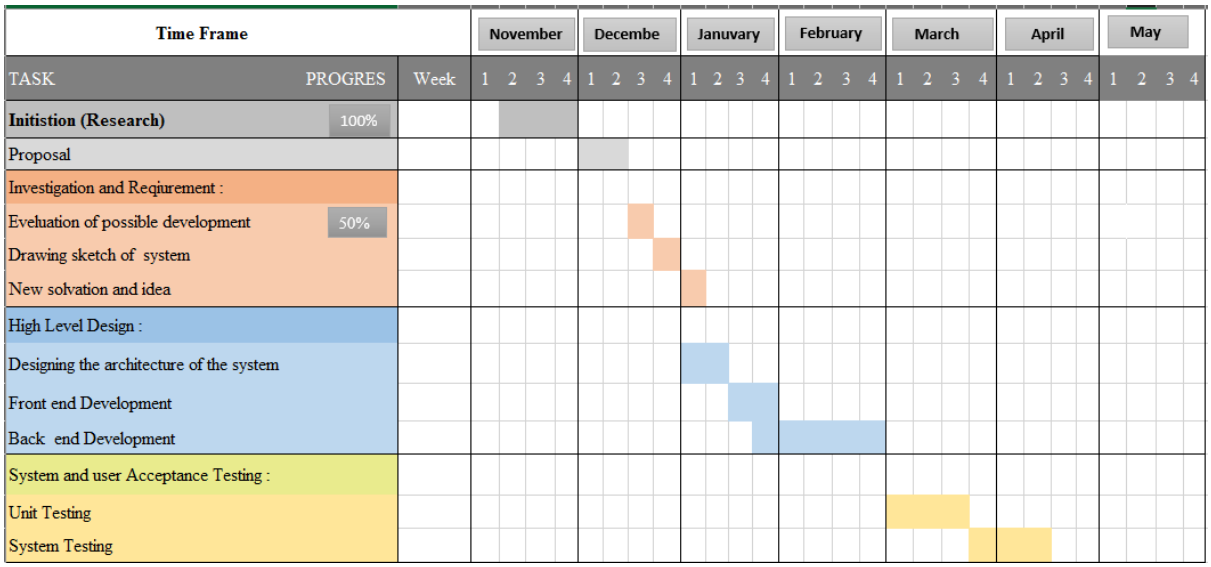


Figure 4: Gantt Chart (Part 1)

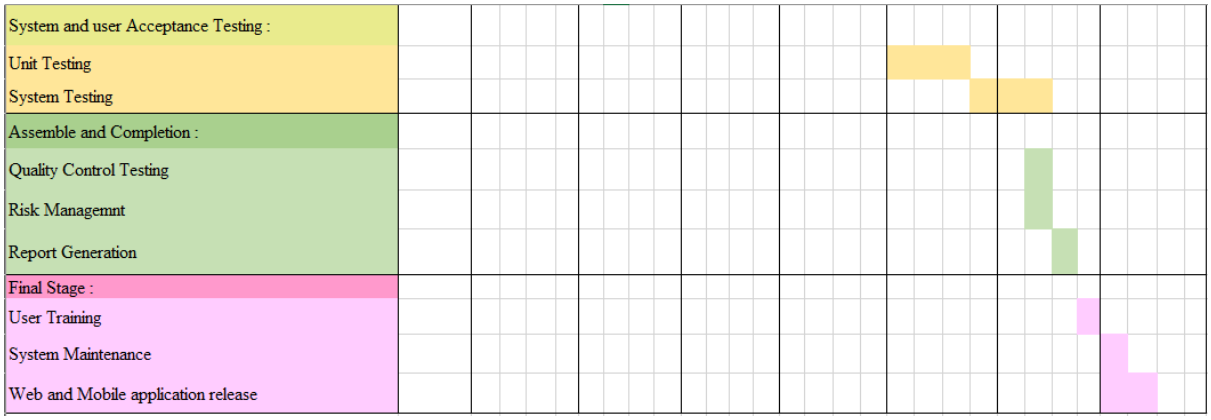


Figure 5: Gantt Chart (Part 2)

Selection of Middleware

Technical Diagram

1.3 Overview of Technical Diagram

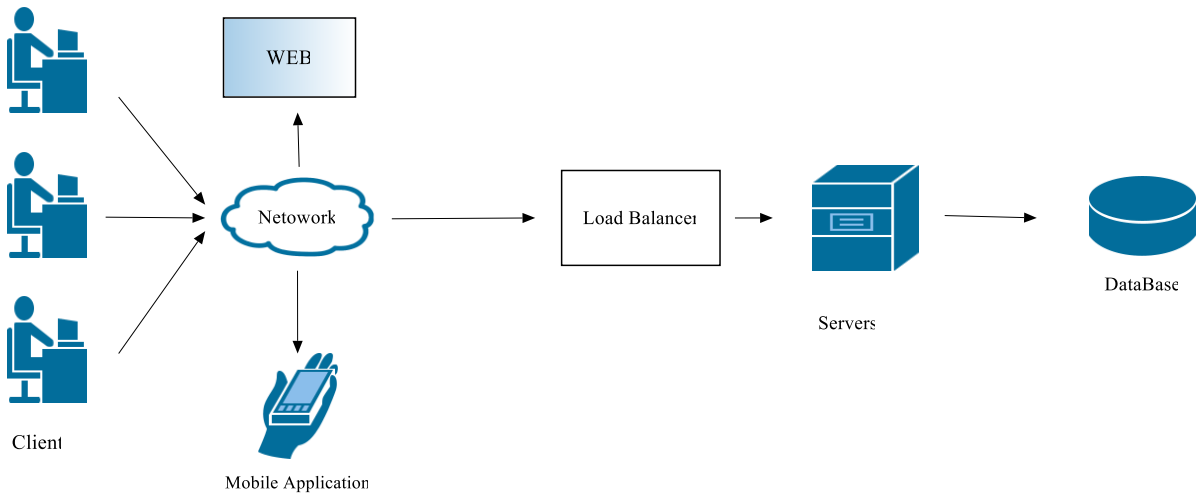


Figure 6: Overview of Technical Diagram

1.4 Technical Architectural Diagram

1.4.1 Overview of Technical Diagram

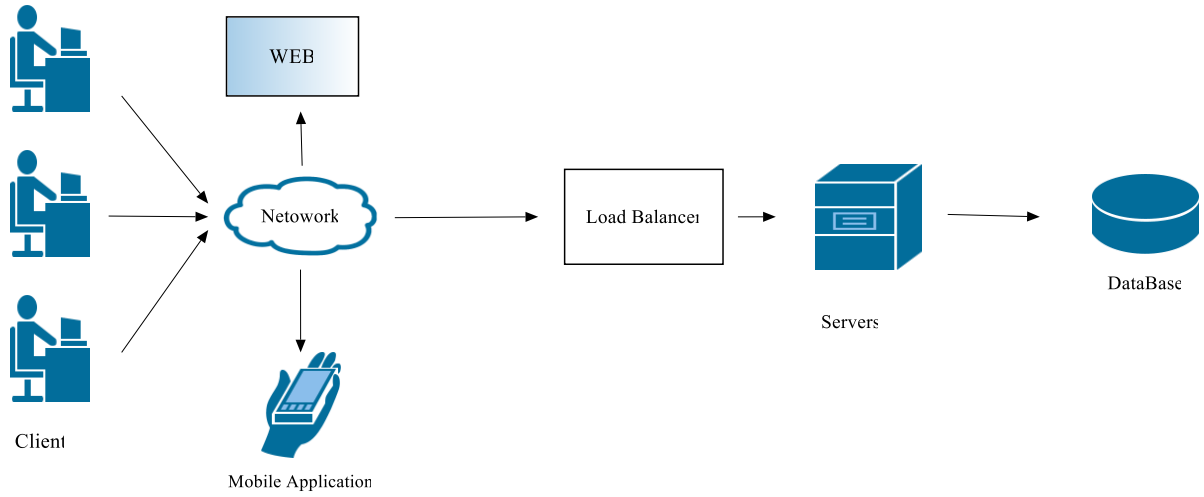


Figure 7: Overview of Technical Diagram

1.4.2 Middleware of Technical Diagram

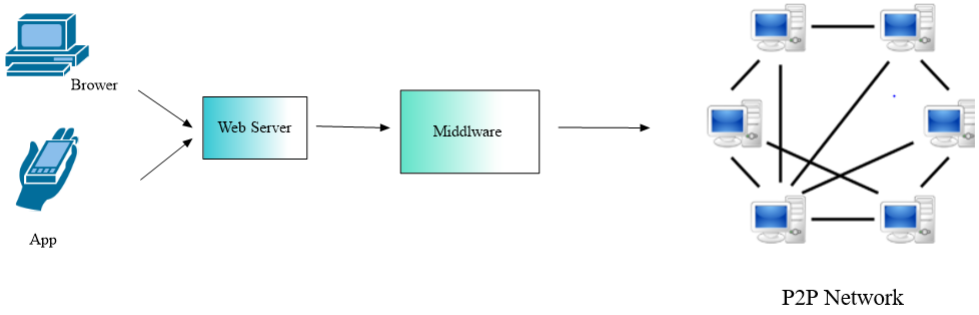


Figure 8: Middleware of Technical Diagram

1.4.3 Technical Architectural Diagram

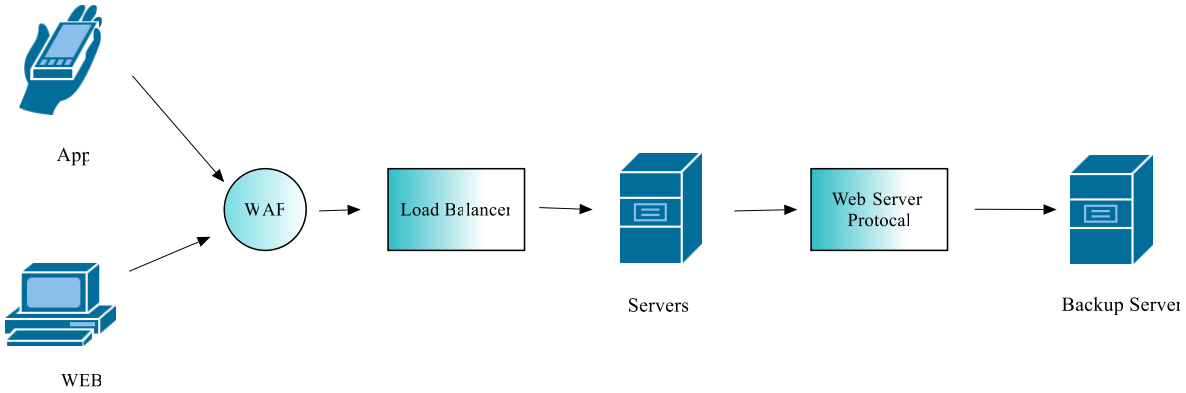


Figure 9: Technical Architectural Diagram

Our system contains a mobile app and a web app, and their total processing is made more efficient by a load balancer, which divides incoming network traffic across a cluster of three dispersed servers.

Load Balancing

A load balancer sits in front of your servers, acting as a "traffic cop," directing client requests across all servers capable of satisfying those requests in a way that maximizes performance and reformulation testing while ensuring that no single server is overworked, potentially degrading performance. The load balancer transfers traffic to the remaining online servers if a single server goes down. When a new server is added to a server group, the load balancer begins sending requests to it automatically.
A load balancer accomplishes the following tasks in this manner:

- Client requests or network load are efficiently distributed among numerous servers.
- Sends requests exclusively to online servers, ensuring high availability and reliability.
- Provides the flexibility to add or remove servers as need dictates.

1.5 With Load Balancing

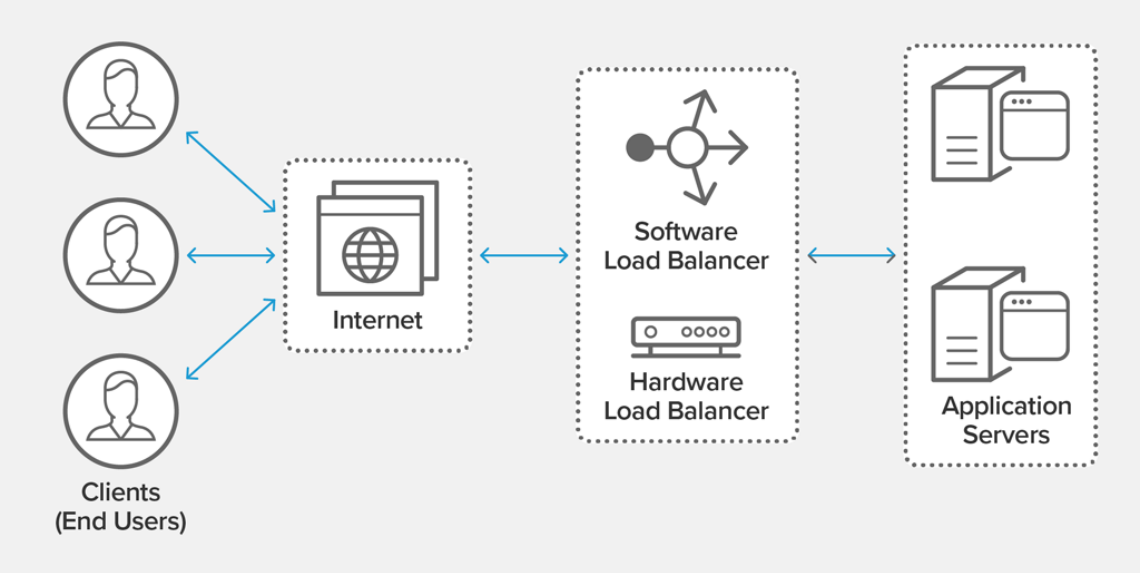


Figure 10: With Load Balancing

1.6 Without Load Balancing

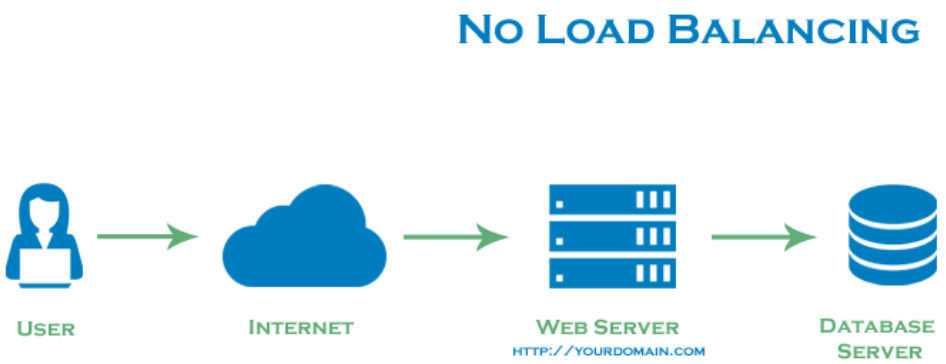


Figure 11: Without Load Balance

Tolerance to Network Failure

As our system run in three servers beside the load balancer, just in case of a network failure it automatically masses to the backup server, thus no network failures can influence the system function.

System architectural Diagram

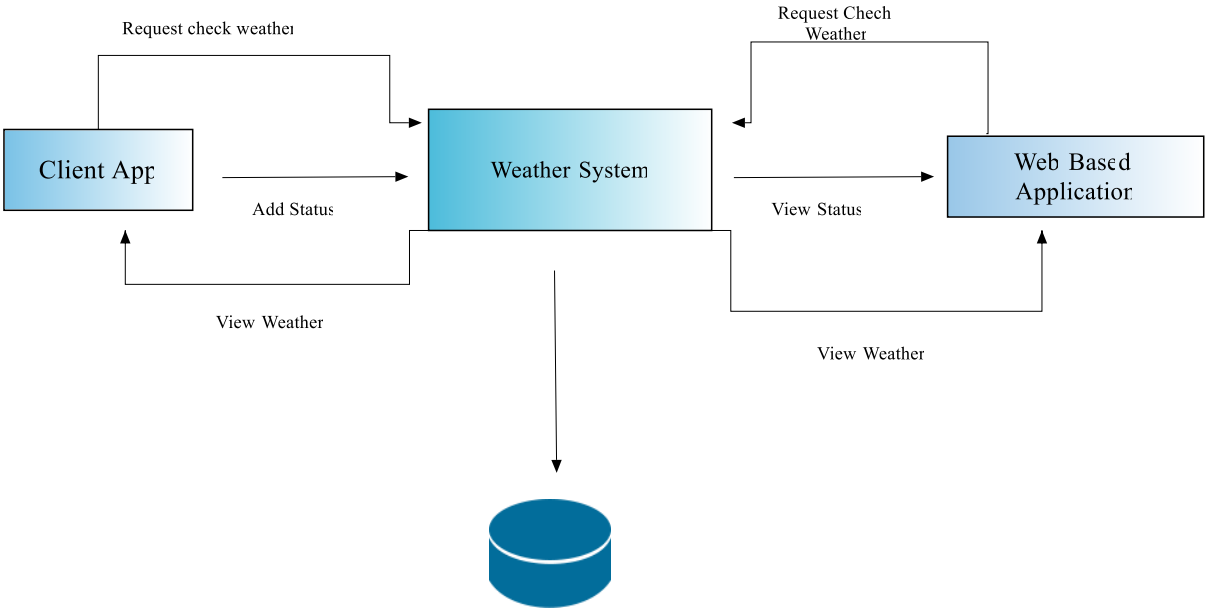


Figure 12: System Architectural Diagram

1.7 User Requirements

The Graphical User Interface system is designed in a way where the user can:

- Users can check weather without having to sign up/sign into the system.
- The users just must enter the city name of the preferred place where they want to know the weather of.
- The weather will be displayed in an instance for the user to view.
- Users also can check posts that are posted by other members of society.

1.8 Functional Requirements –

- We tend to run multiple copies of each element on different machines.
- Data is replicated in order to avoid single machine failure. In this way, loss of data can be avoided.
- The system runs elements on separate machines.
- Has redundancy at all layers.

API / Middleware Justification

Initially when we run the system, we have used middleware as the “glue” holding the applications made together, with this “glue” we can connect the three applications together.
For the implementation of the middleware Laravel was used.
Functions done by the middleware:

- Authenticates the user.
- Executes code.

Software components used

- Laravel – Laravel was used to create the API for the system, this is a free, open-source web framework that is used for web application development. It follows a model-view-controller design pattern.
- MySQL – Open-source relational database management system, it uses Structured Query Language, one of the most famous languages for managing content in a database.
- Android Studio – Android Studio is a mobile application development platform; Android Studio has their own emulators which makes it easy for developers to use and maintain. Android studio has fast execution, which provides time effectives for developers.
- NetBeans – integrated development environment for Java. NetBeans was used to create our stand-alone application.

Issues faced and actions taken

Issues Faced	Actions Taken
Software Installation Failure	With necessary research by following tutorial and LinkedIn courses, we manage to resolve the issues that had materialized.
Connecting to API	
"PHP artisan migrate" did not migrate	

Table 2: Issues faced, and actions taken

Development phases

Development phases	Description
1. Planning	When it comes to defining a project's boundaries, it is mandatory to consider the potentiality of how largesse the scope can be. By composing a project plan, a calculation of the spatial sense (timeliness) of the project will be incited. This will include a breakdown of tasks, a conclusion of the inherent deadlines and overall time taken to complete the tasks. All necessary requirements are accumulated to determine the sequencing the work, including a creation of an initial project schedule.
2. Analysis	The analysis stage will decipher the system's critical features and requirements. This will establish whether it is worthwhile to develop. It will also determine a requirement of the possible tools and technologies to be utilized, a potential project schedule, necessary budget estimations, etc.
3. Designing	Designing will aid in roughly determining the shape and form of what must be assembled. It will be utilized in the form of a template. This will dictate the implementation proceedings, software architecture, database, and frontend designs.

4. Coding and debugging	Frontend development, Database development, Website creation, and the implementation and development of backend logical coding has been contrived. Correspondingly, a Mobile Application with implemented logical coding and a frontend is generated. In addition, an API to interface and a MySQL database has been created.
5. Testing	Functional testing is executed to see if we are achieving the expected and needed outcomes. All code units in the project have been tested, closely following the unit testing process. Integration Testing has been conducted after Once the system's modules were integrated, the full system was effectuated with end-to-end testing. We have also exercised User Interface Testing. This process would entail checking whether the UIs have been designed in fulfilment of the system's specifications. Furthermore, Load Testing has been conducted within the confines of non-functional testing to emphasise the system's overall quality.
6. Deliver the product	In the final development stage, the website, mobile application, and report is completed and presented.

Table 3: Development Phases

Tasks undertaken

1.9 Database

Shown below are the Database tables which contain important info, such as storing users' data and other information, post information and city data.

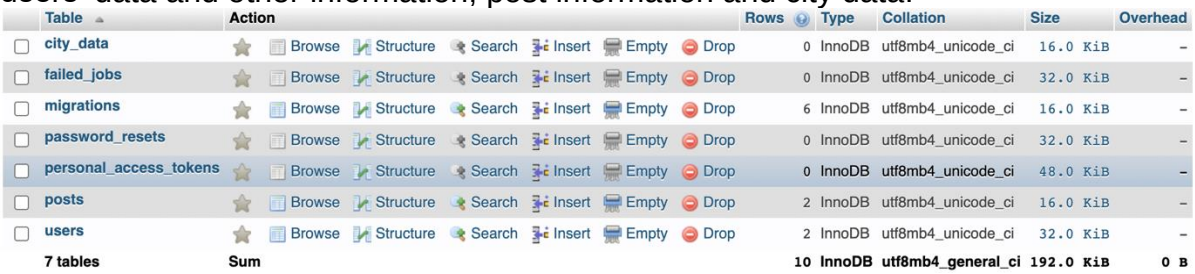


Table	Action	Rows	Type	Collation	Size	Overhead
city_data	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
posts	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
7 tables	Sum	10	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B

Figure 13: Database

User's data is stored as follows.

	id	name	email	email_verified_at	password	remember_token	created_at	updated
Edit Copy Delete	1	test	test@gmail.com	NULL	$2y$10$i8os5MQHfKec.zTEqhiHTu9yG89IPBEtaiI2wbp3AOg...	NULL	2022-05-08 17:30:28	2022-05-
Edit Copy Delete	2	Nilhan	neo@gmail.com	NULL	$2y$10$Dfcex38ZjwhoU5q3AQ4qKe9WvkcWY2dsdl4melniipA...	NULL	2022-05-08 19:39:21	2022-05-

Figure 14: Stored data of users

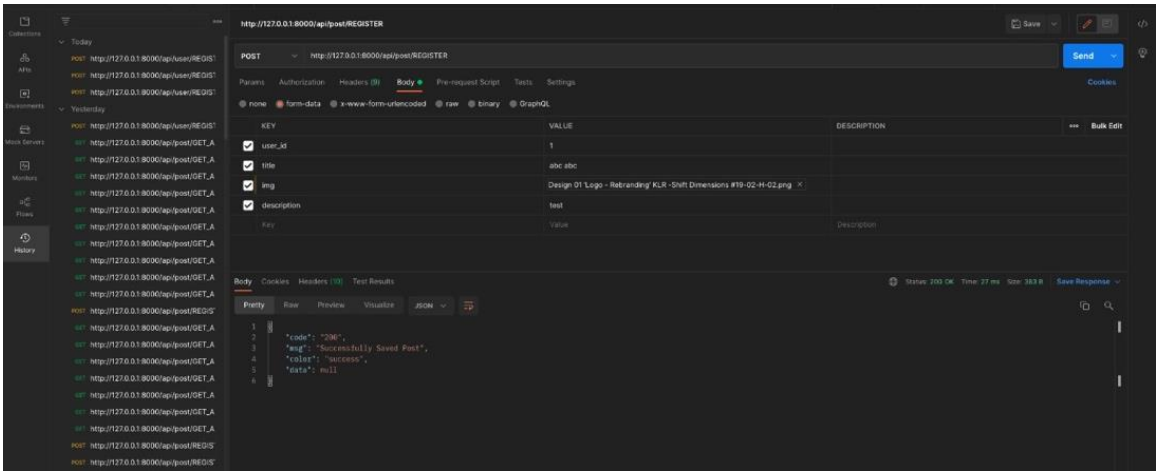
Posts submitted by users are stored in here.

+ Options

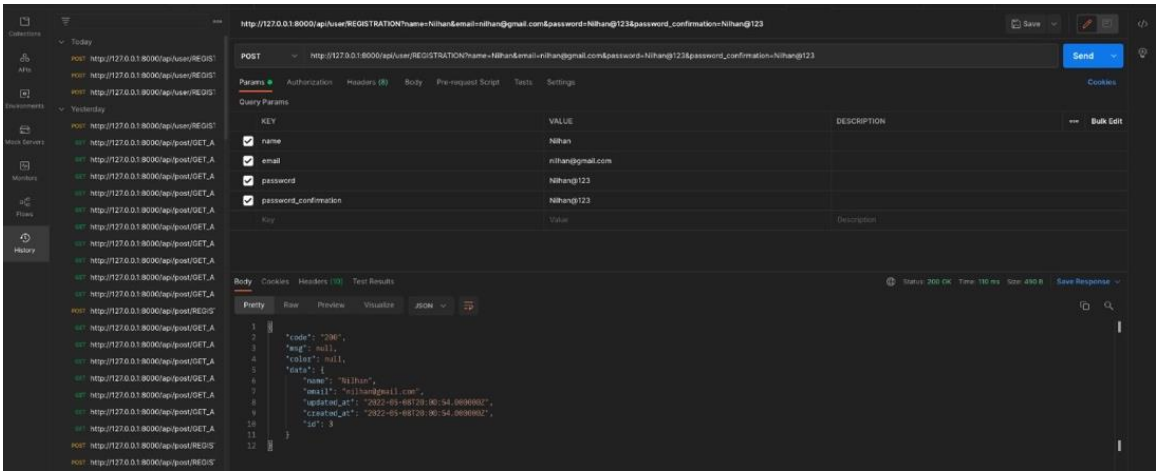
	id	user_id	title	img_url	description	status	created_at	updated_at
Edit Copy Delete	1	1	test post		test post desc	1	NULL	NULL
Edit Copy Delete	2	1	test title	1652032994.png	data	1	2022-05-08 18:03:14	2022-05-08 18:03:14

Figure 15: Post submission

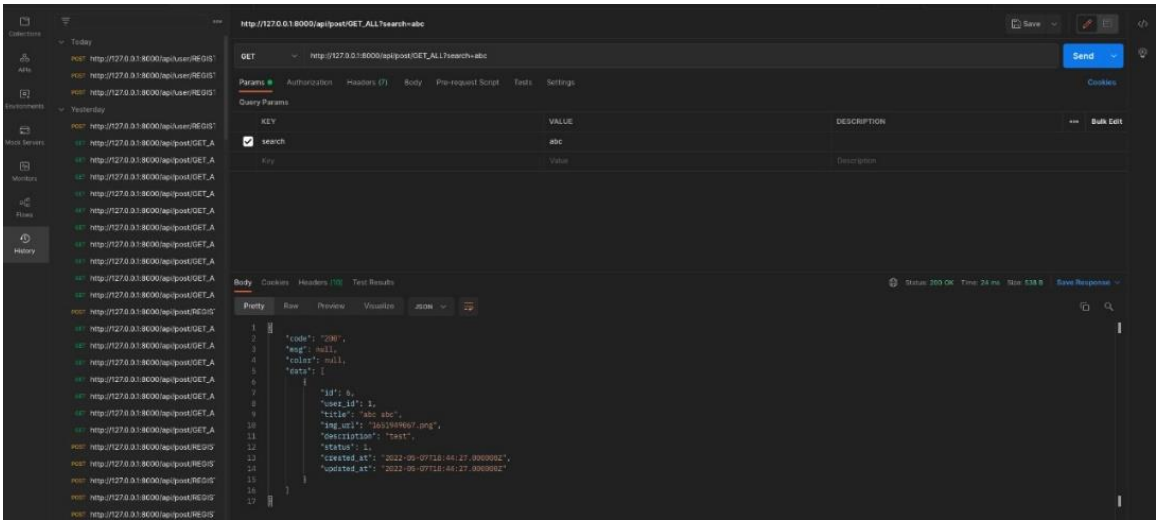
1.10 Postman API requests and testing



1.10.1 Add Post Request

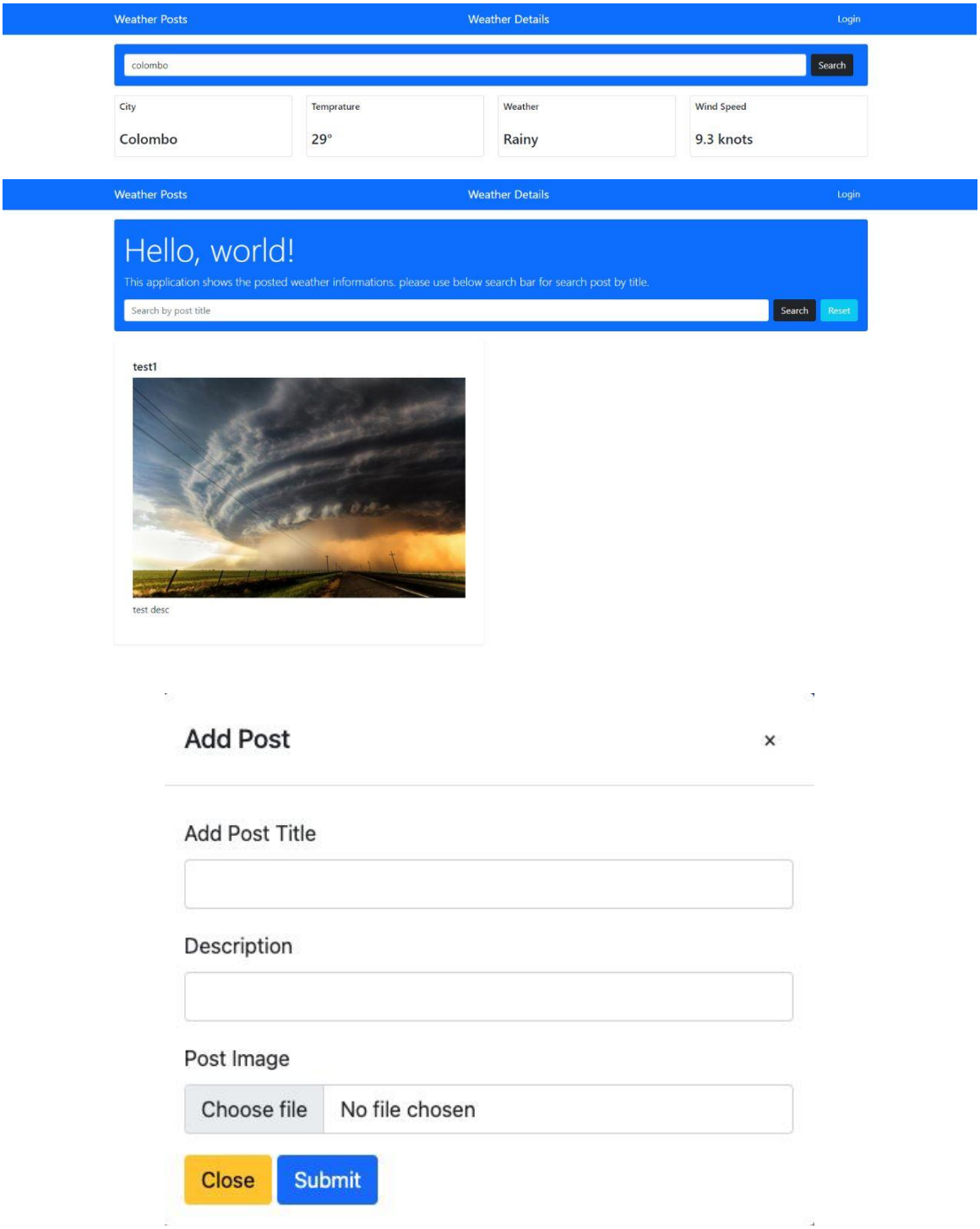


1.10.2 Login Request



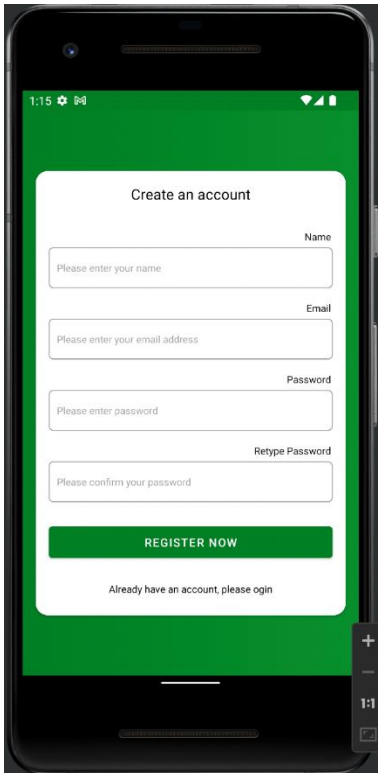
1.11 Screenshots of the website

The web application as shown below is the interface the user is greeted with when

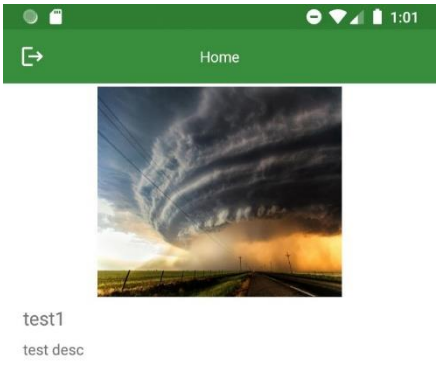


1.12 Screenshots of the Mobile Application

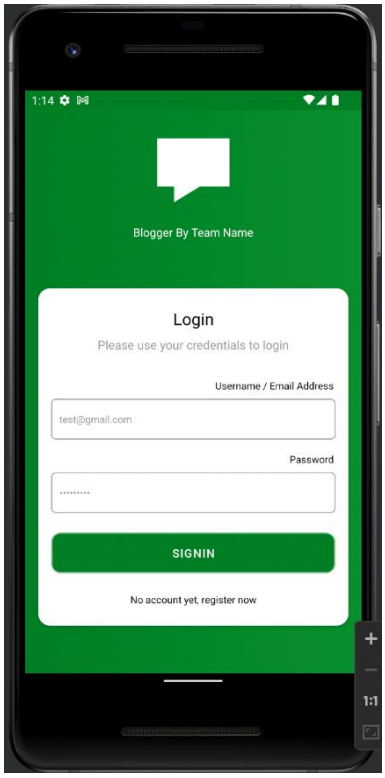
This is the interface the client is greeted with when he/ she is in need of creating an account to gain privileges of posting updates on the weather post page on the web page of this distributed application.



After the user creates an account, he or she can login through the app to see the weather-related posts that have been submitted as shown below



This is the user login page where he or she enters their email and password to login to the app to view the posts.



1.13 Main codes of the website

1.14 Main codes of the mobile application

Main activity is where the code responsible for the temporary splash screen is visible as you access the mobile application.

1.14.1 Create an account mobile application code

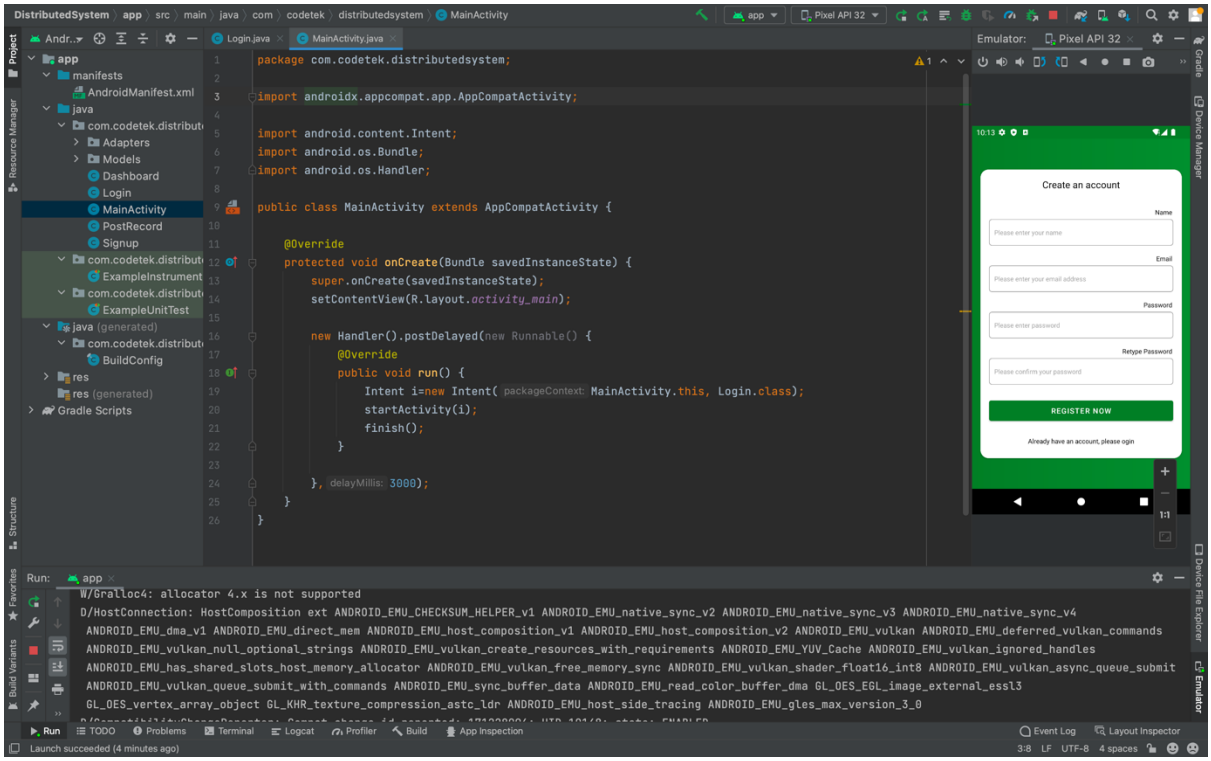


Figure 16: Create an account

1.14.2 Login

Login screen code is present in this code view as shown with the running interface in the right side of the screen shot.

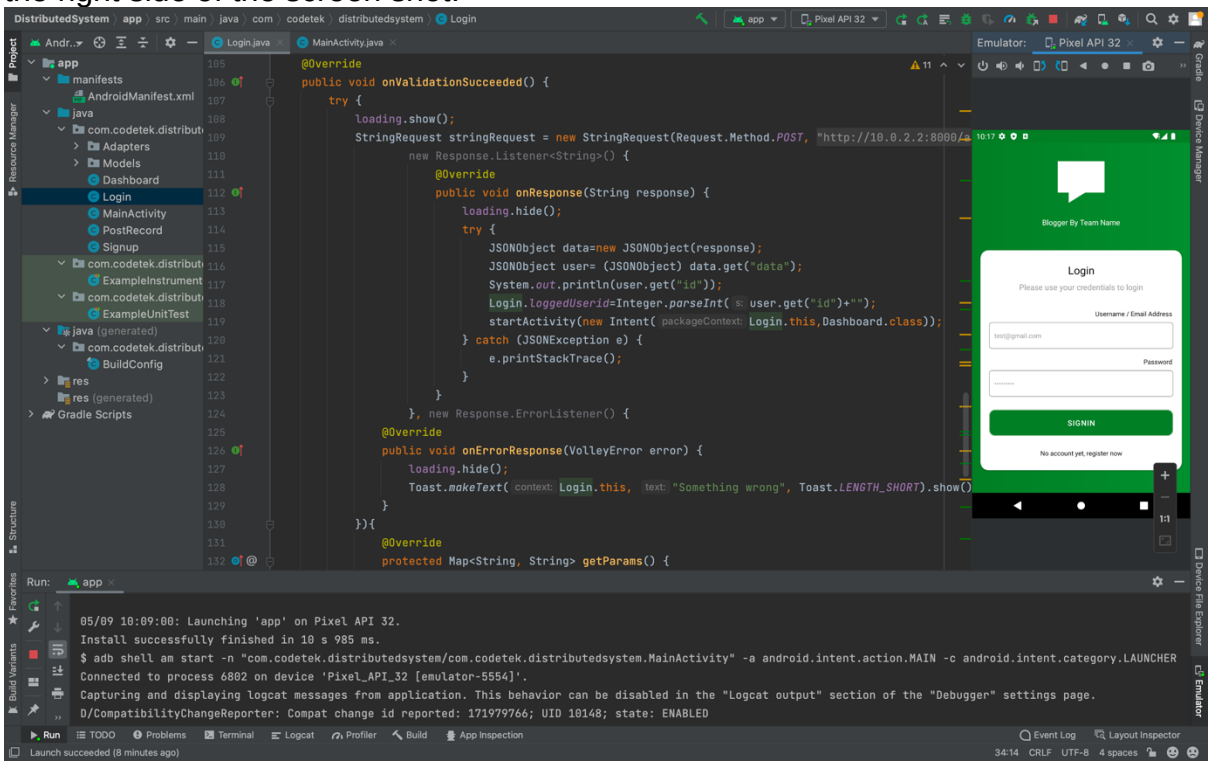


Figure 17: Login code

1.14.3 Dashboard

This shows the code responsible for the post page in the mobile view where the user can view the posts.

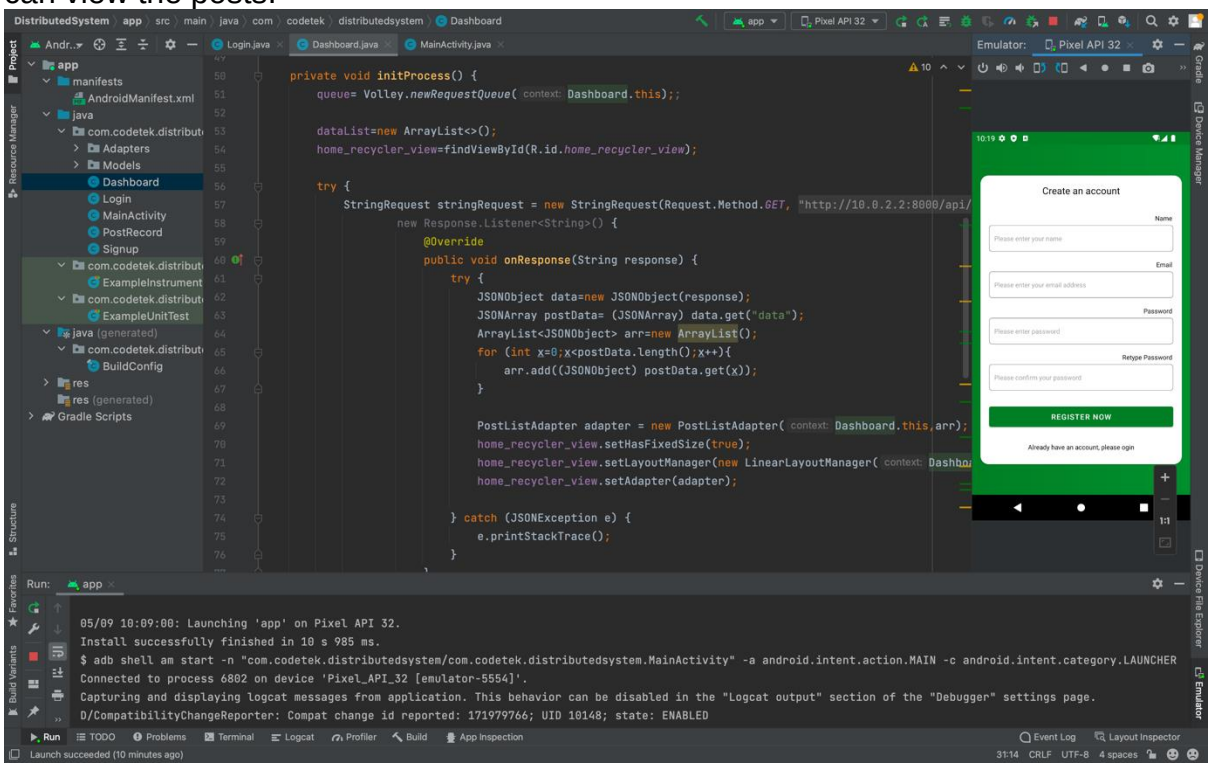


Figure 18: Dashboard Code

1.14.4 Post records

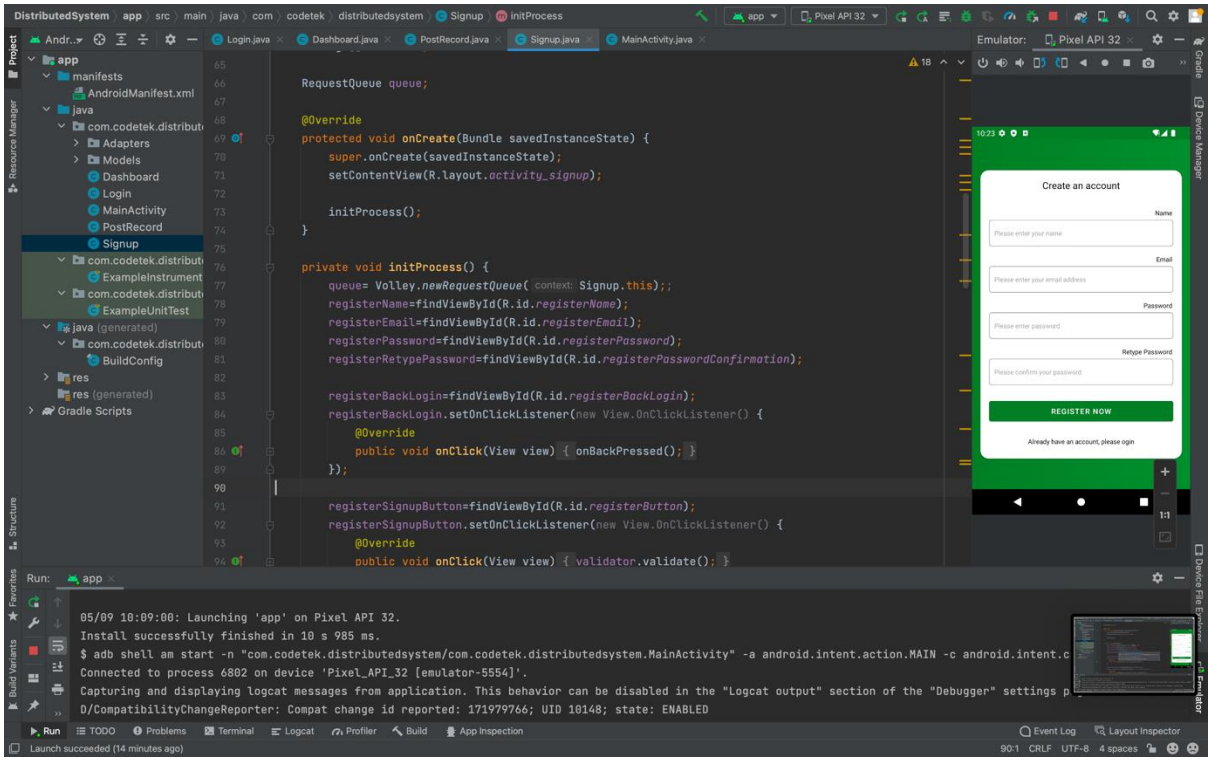


Figure 19: Post Records

1.14.5 Signup Screen

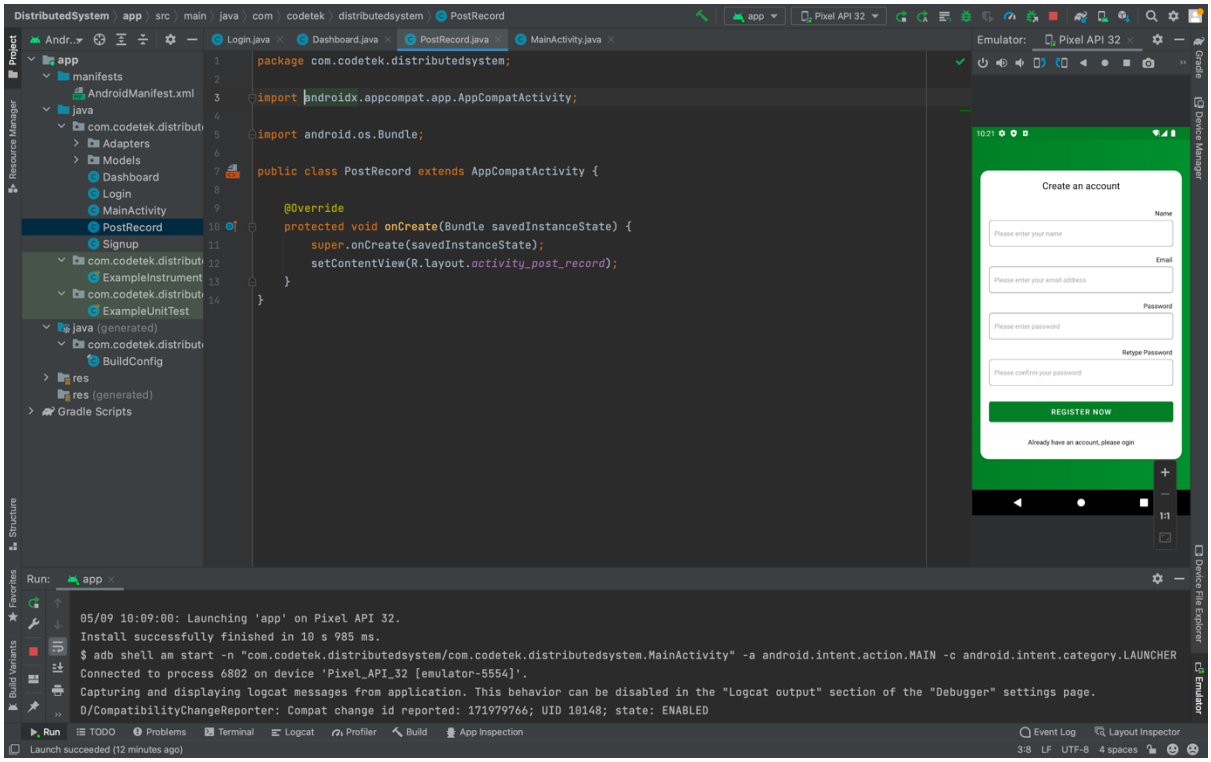


Figure 20: Sign up screen page

1.14.6 Models Folder

The model folder is responsible for storing user data and serialization of that data.

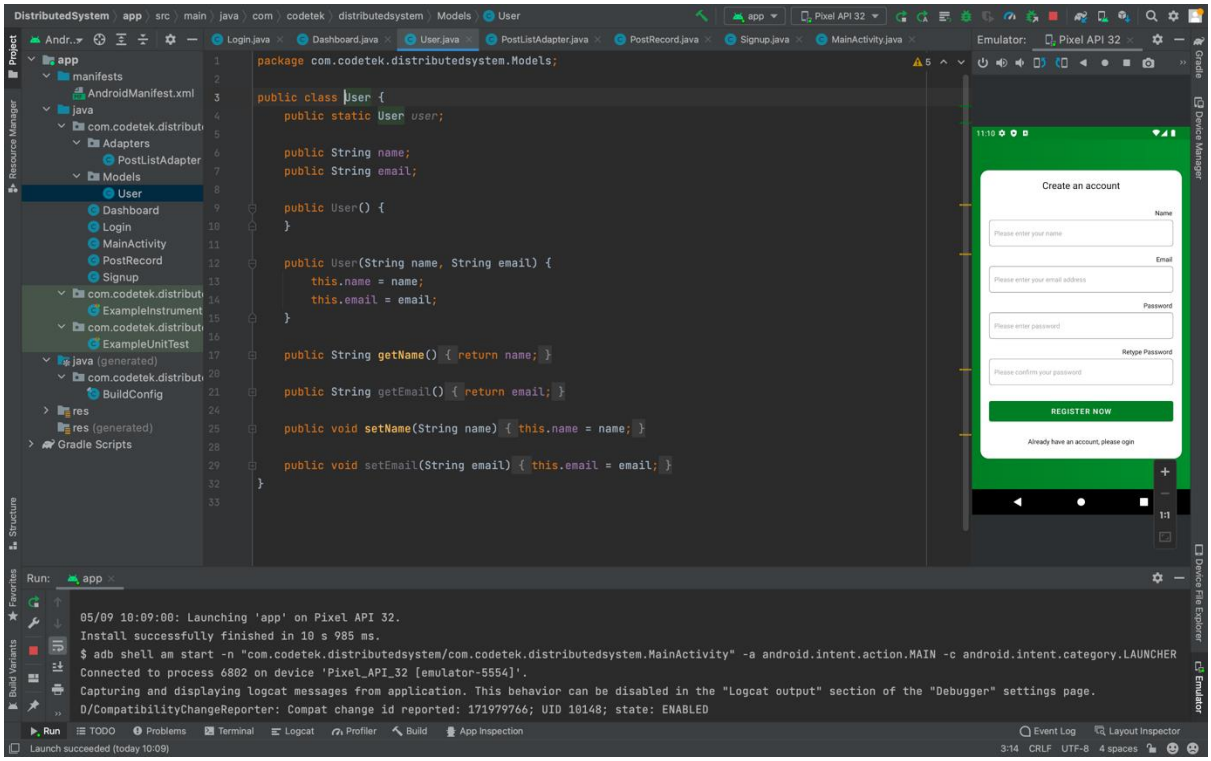


Figure 21: Model Folder

1.14.7 Post list adapter folder

This block of code is responsible for the way on which the posts are displayed for the users of the application by accordingly segregating the title of post, description, and uploaded images etc.

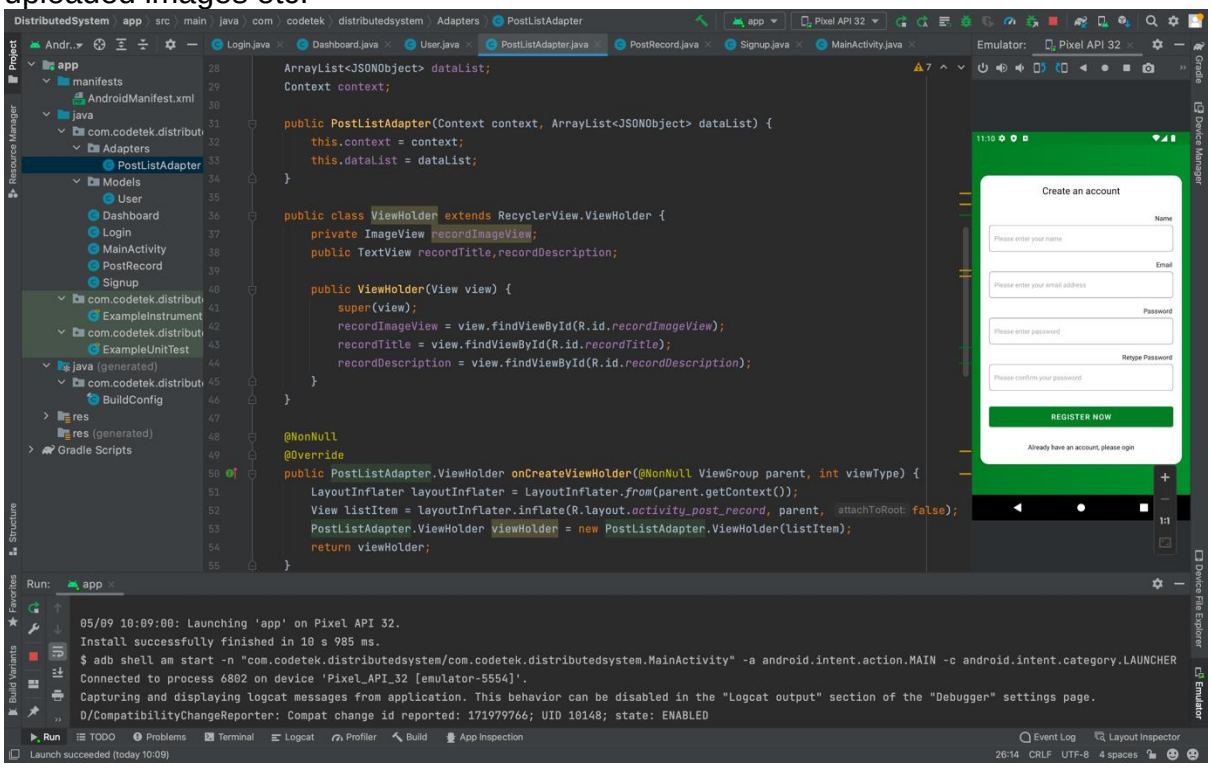


Figure 22: Post list adapter folder

1.15 Main codes of the Desktop application

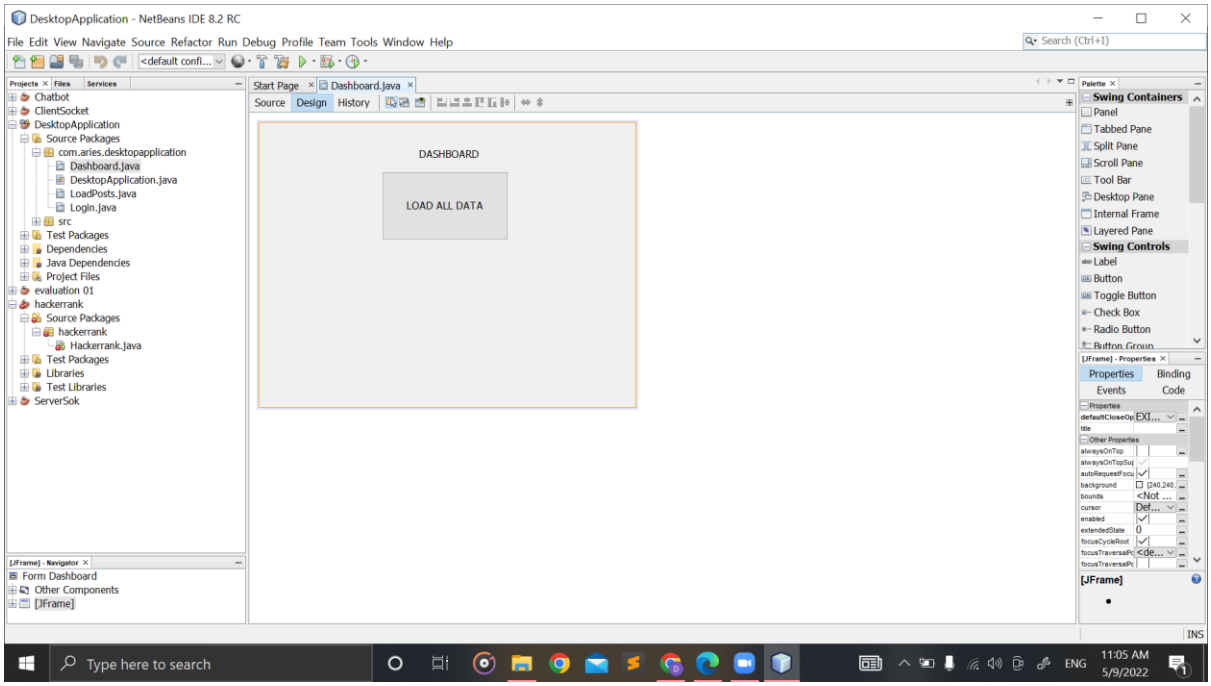


Figure 23: Dashboard

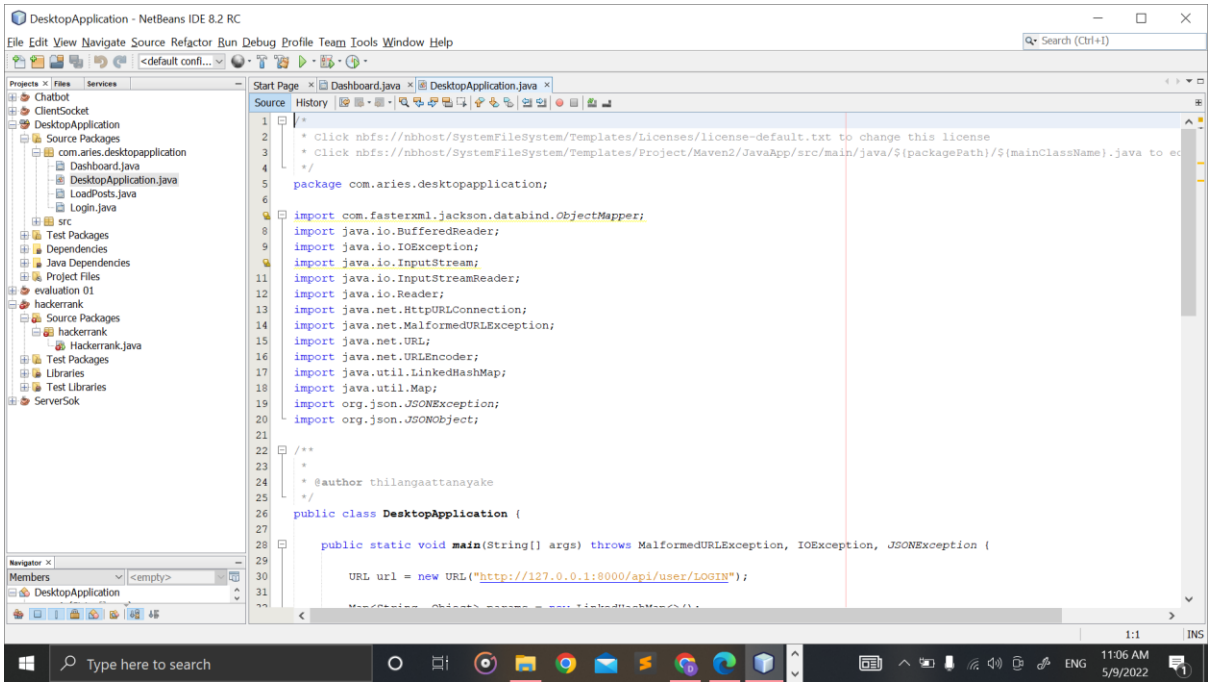


Figure 24: DesktopApplication.java part 1

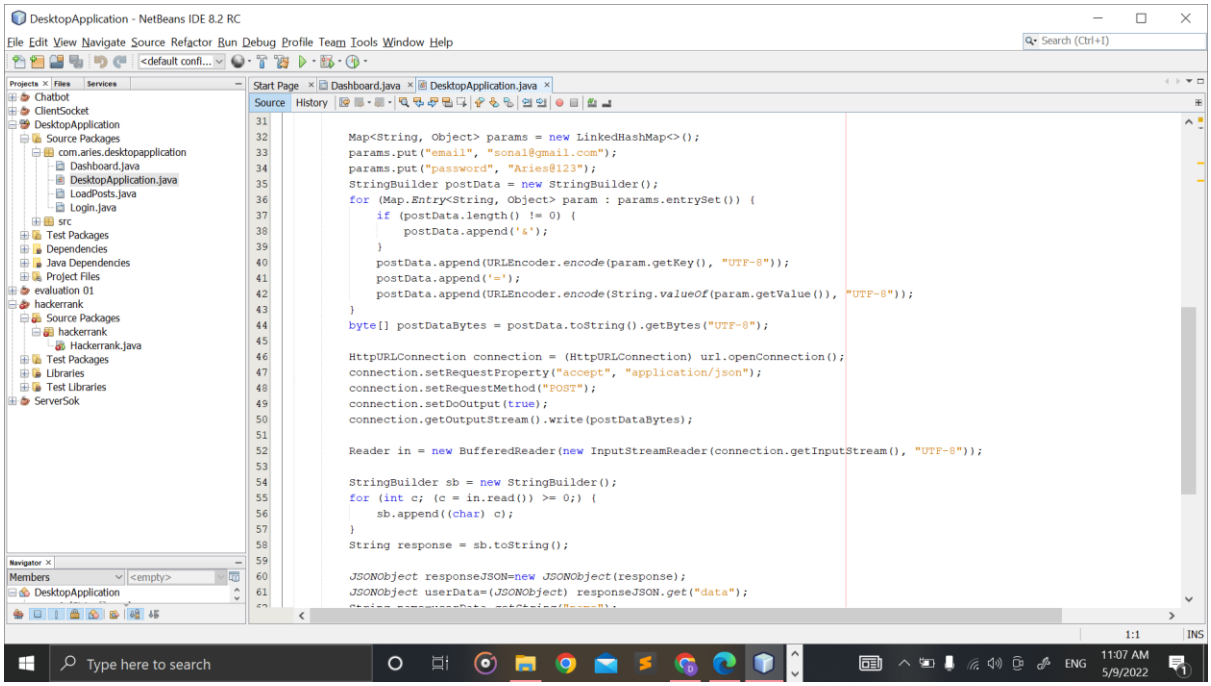


Figure 25: DesktopApplication.java part 2

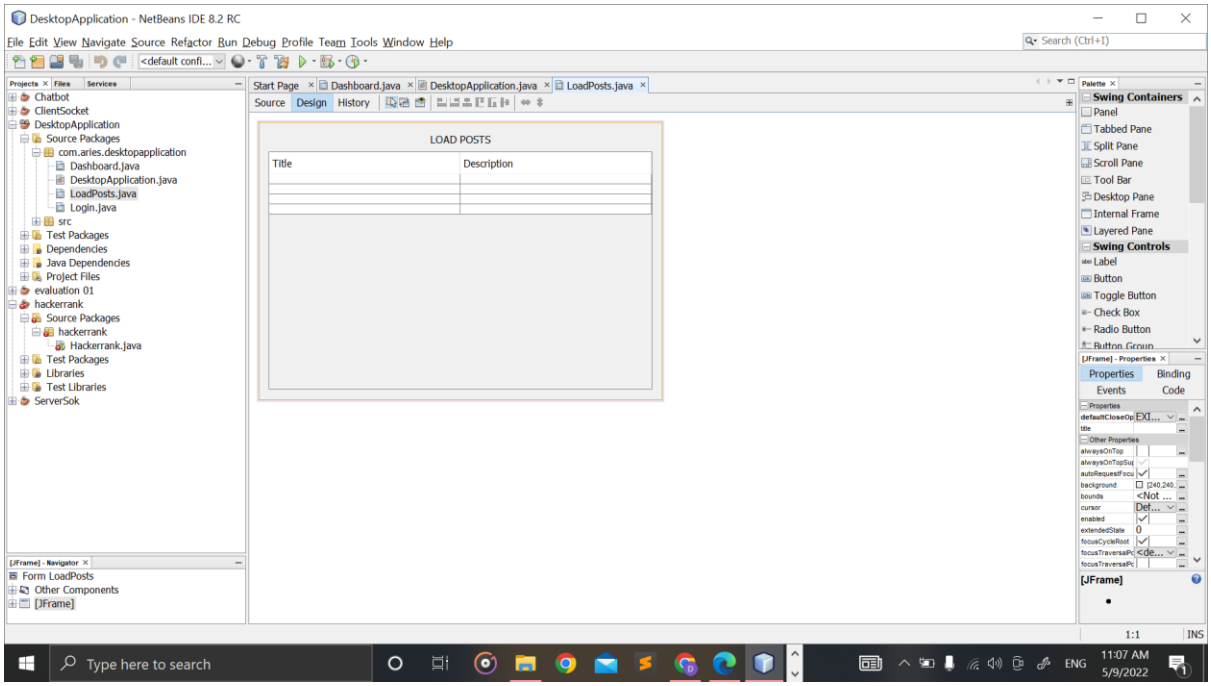


Figure 26: LoadPosts.java

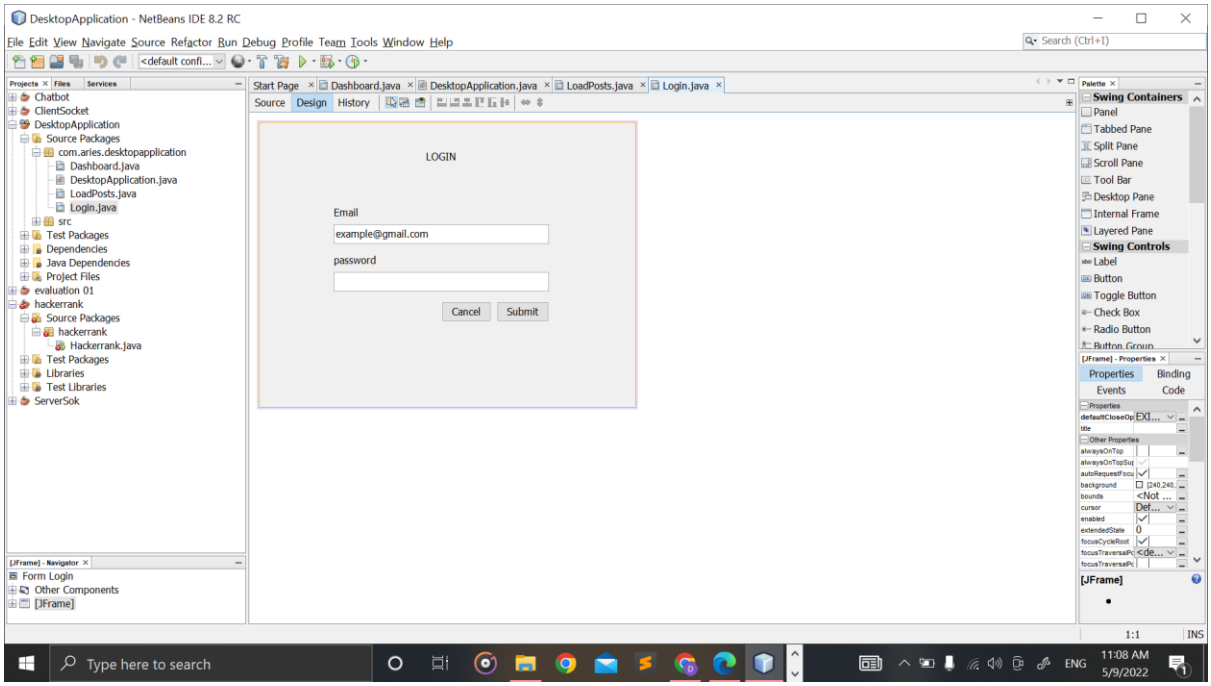


Figure 27: Login.java

1.16 Main codes of the API

1.16.1 Controller

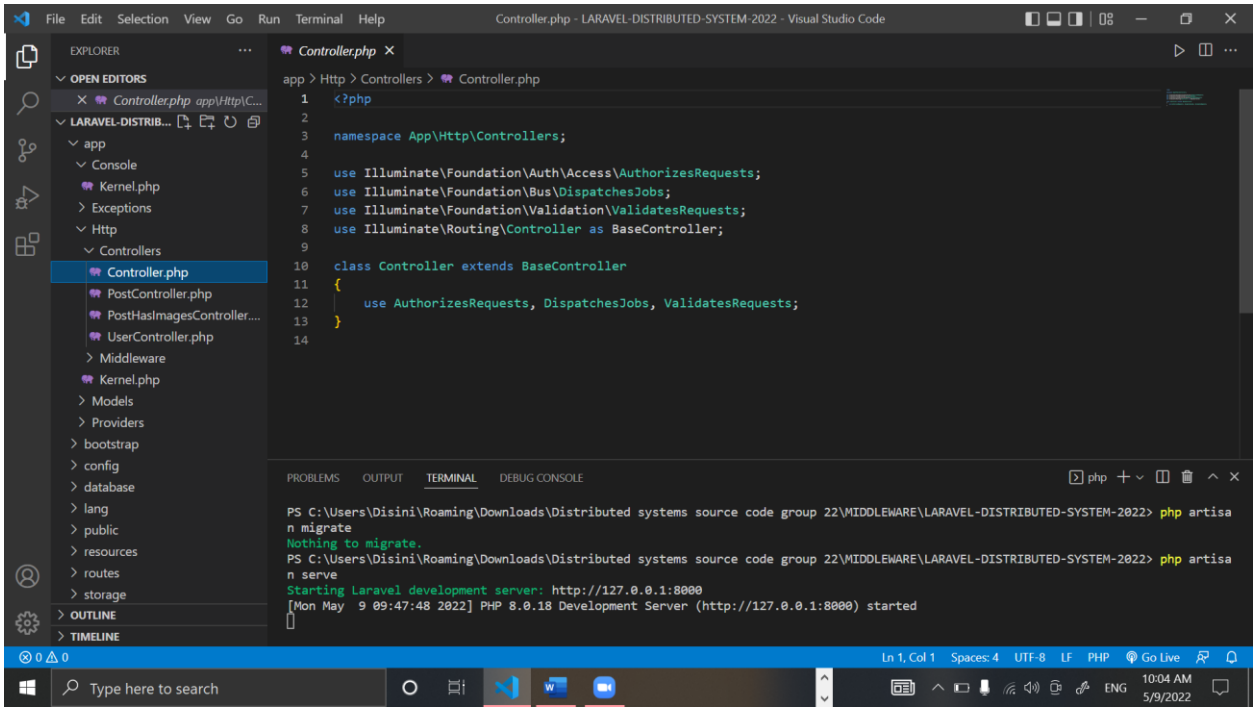


Figure 28: Controller.php

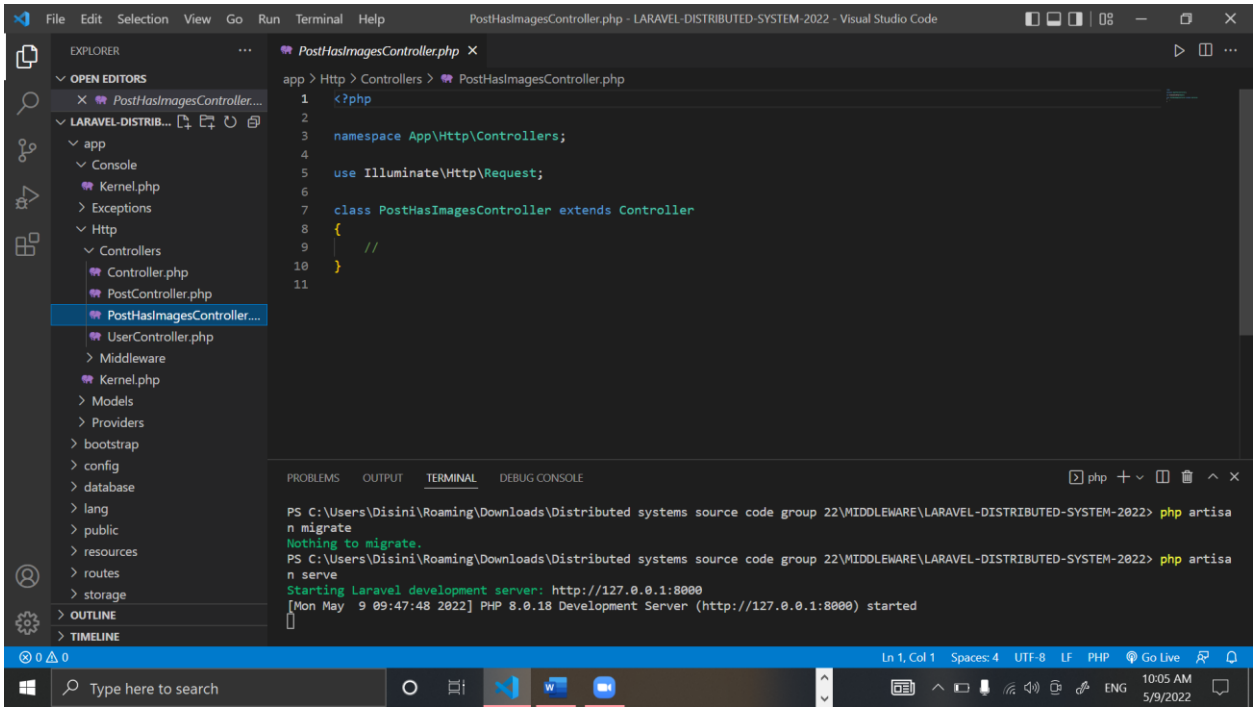


Figure 29: PostHasImagesController.php

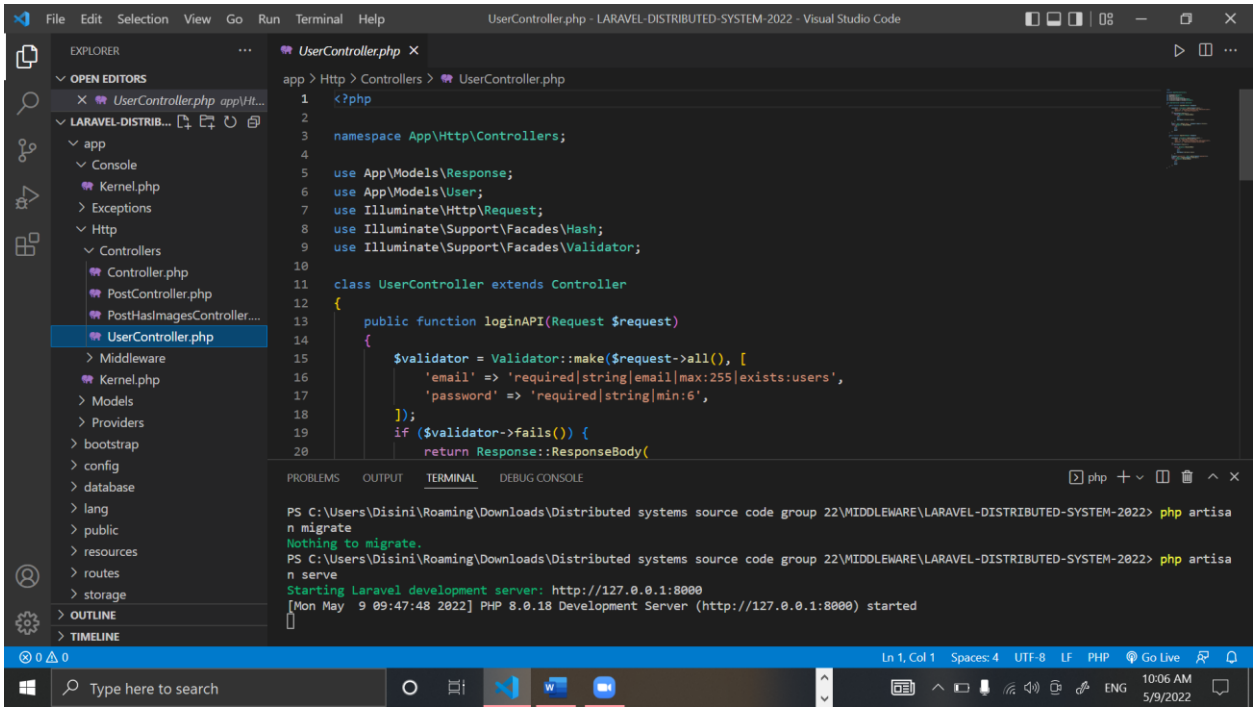


Figure 30: UserController.php part 1

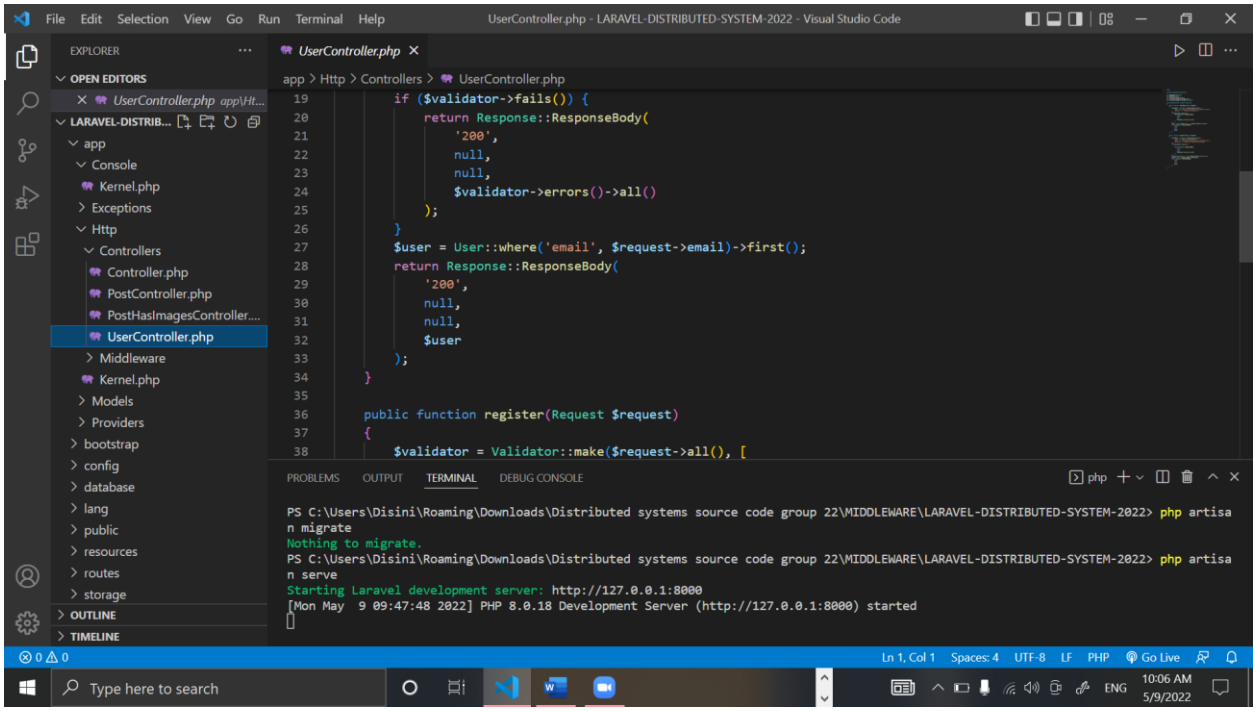


Figure 31: UserController.php part 2

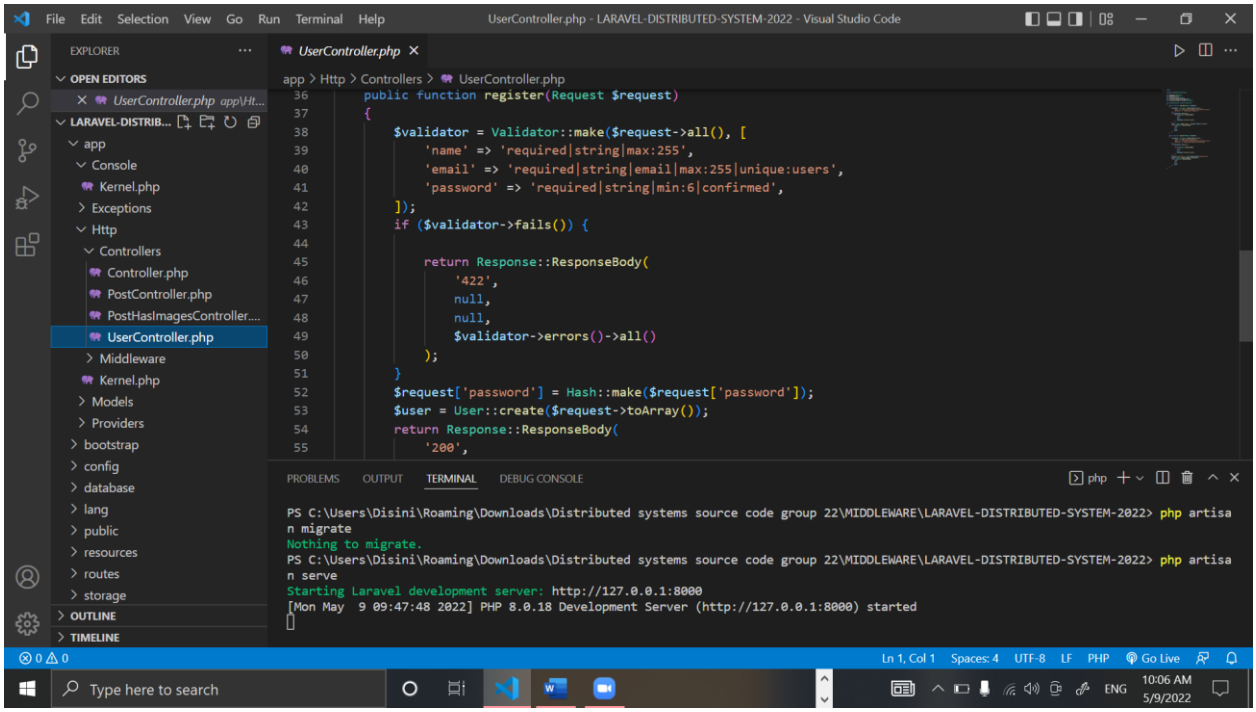


Figure 32: UserController.php part 3

1.16.2 APIs

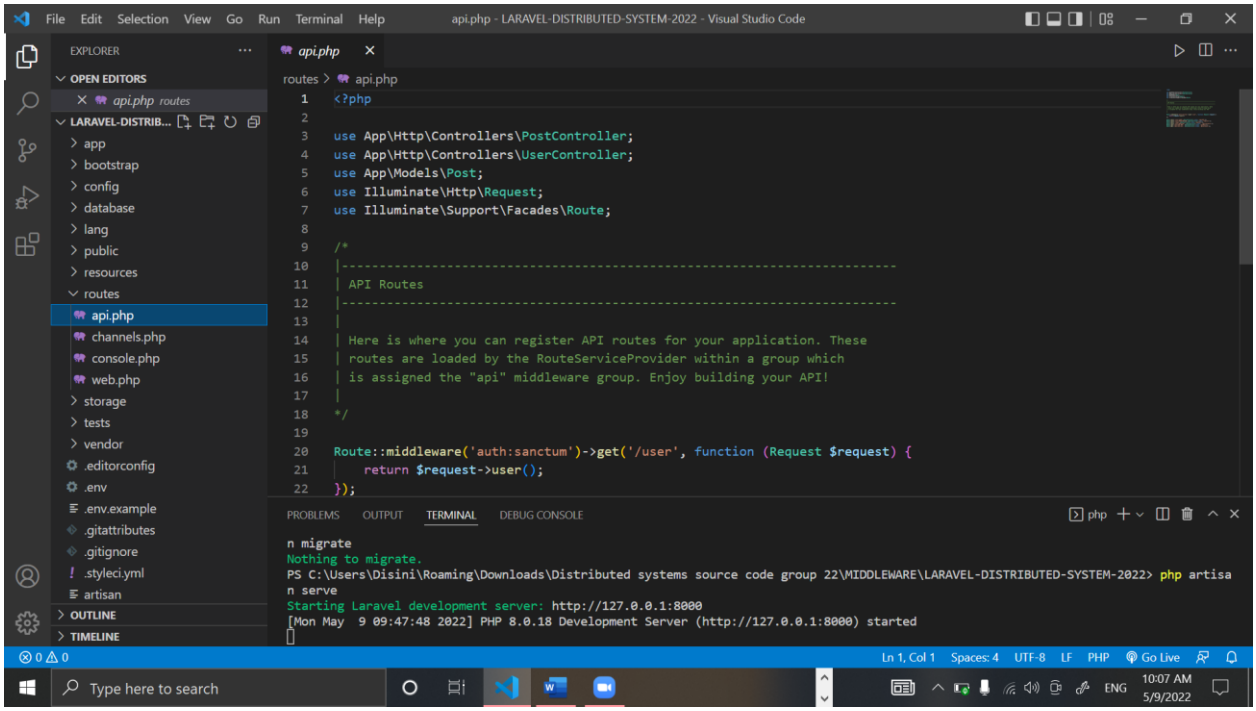


Figure 33: Api.php part 1

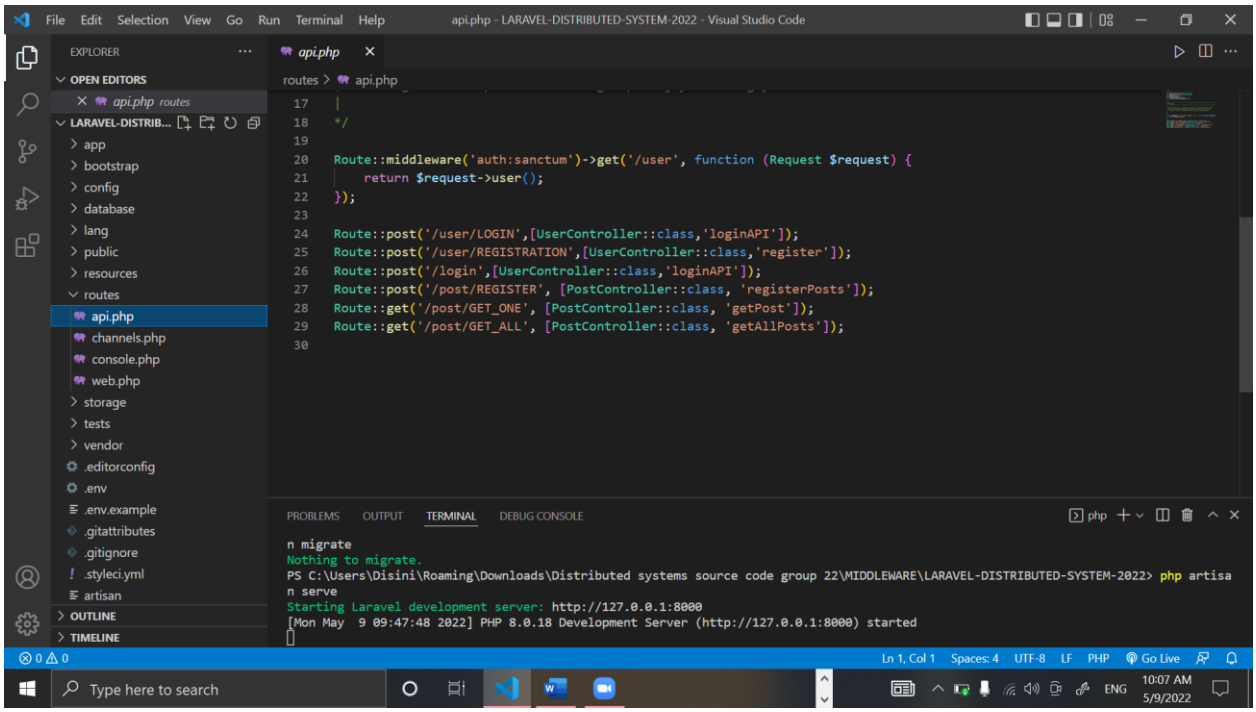


Figure 34: Api.php part 2

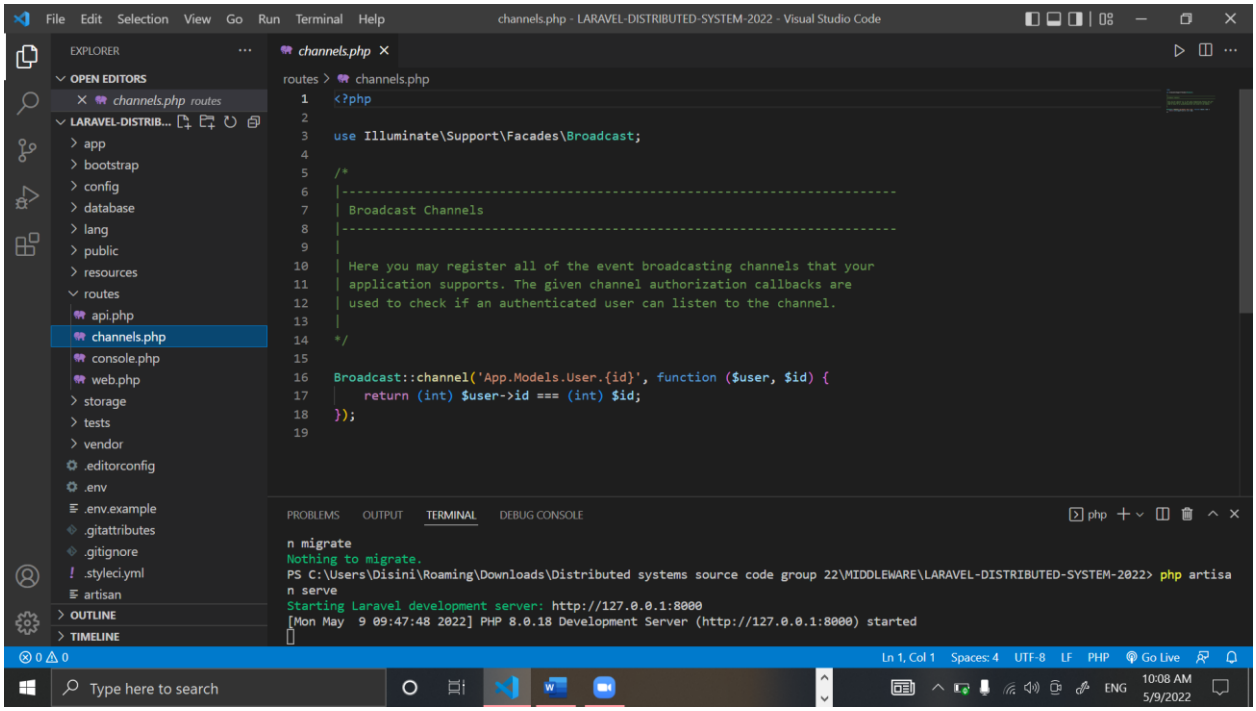


Figure 35: Channels.php

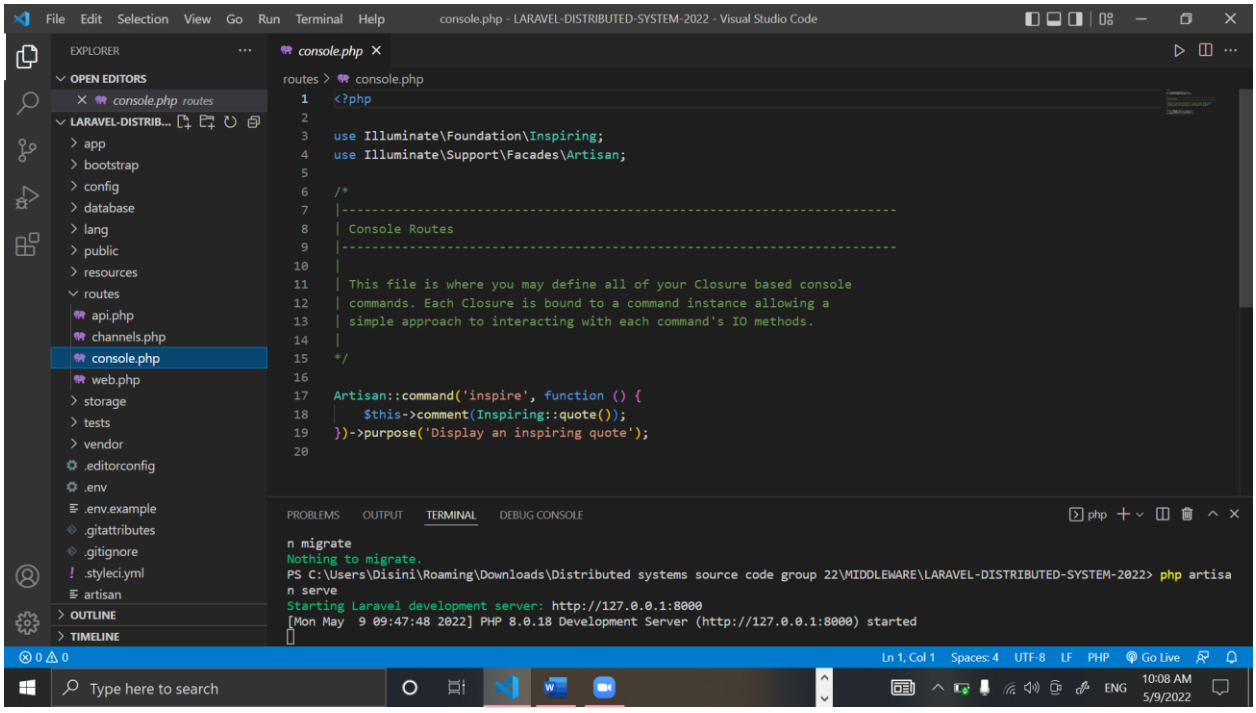


Figure 36: Console.php

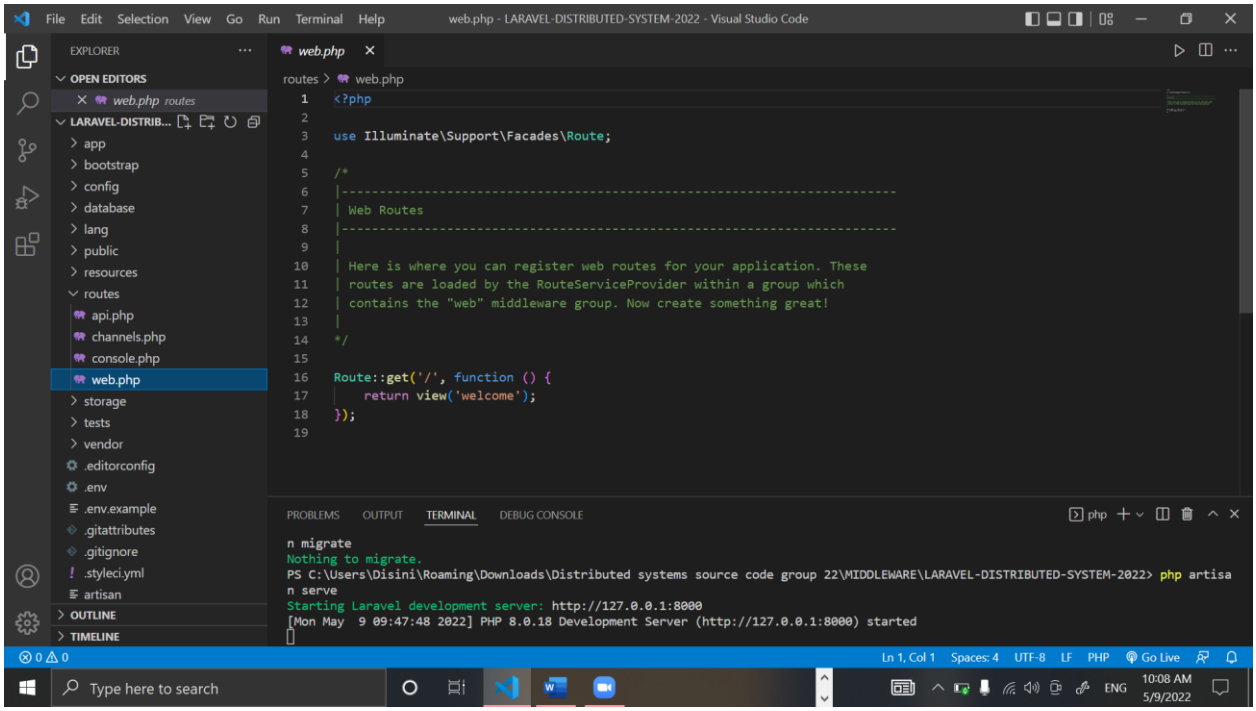


Figure 37: Web.php

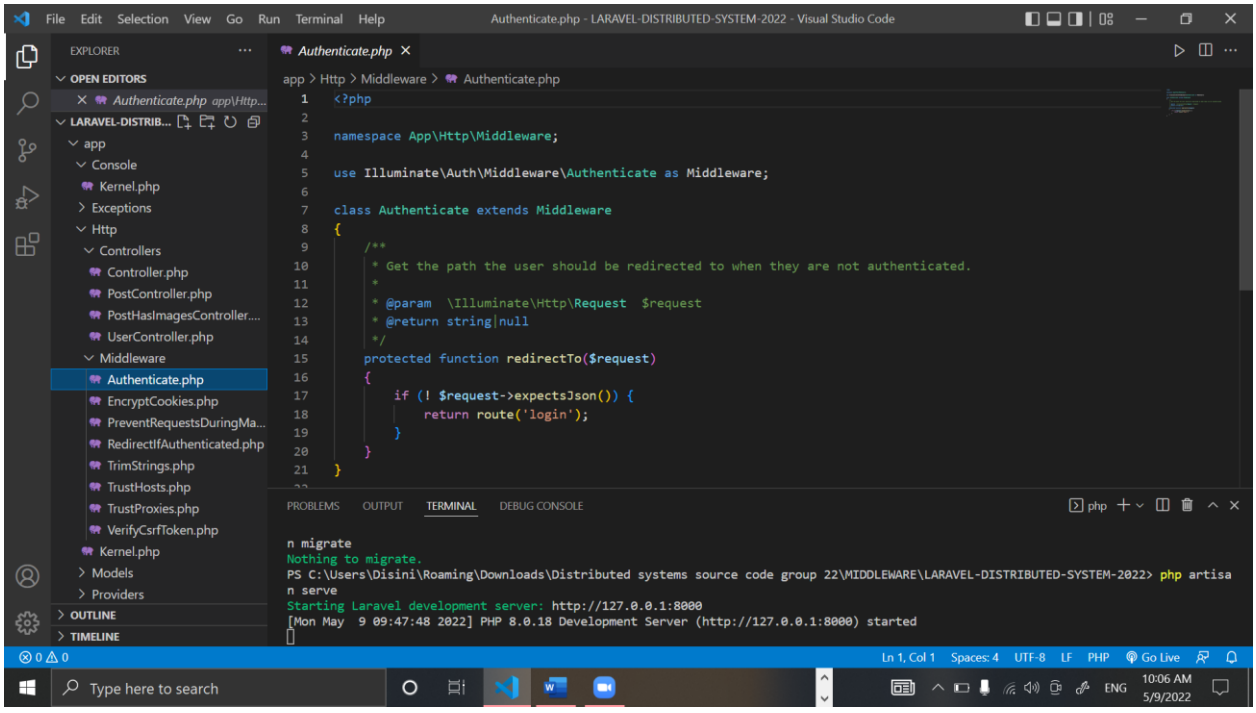


Figure 38: Authenticate.php

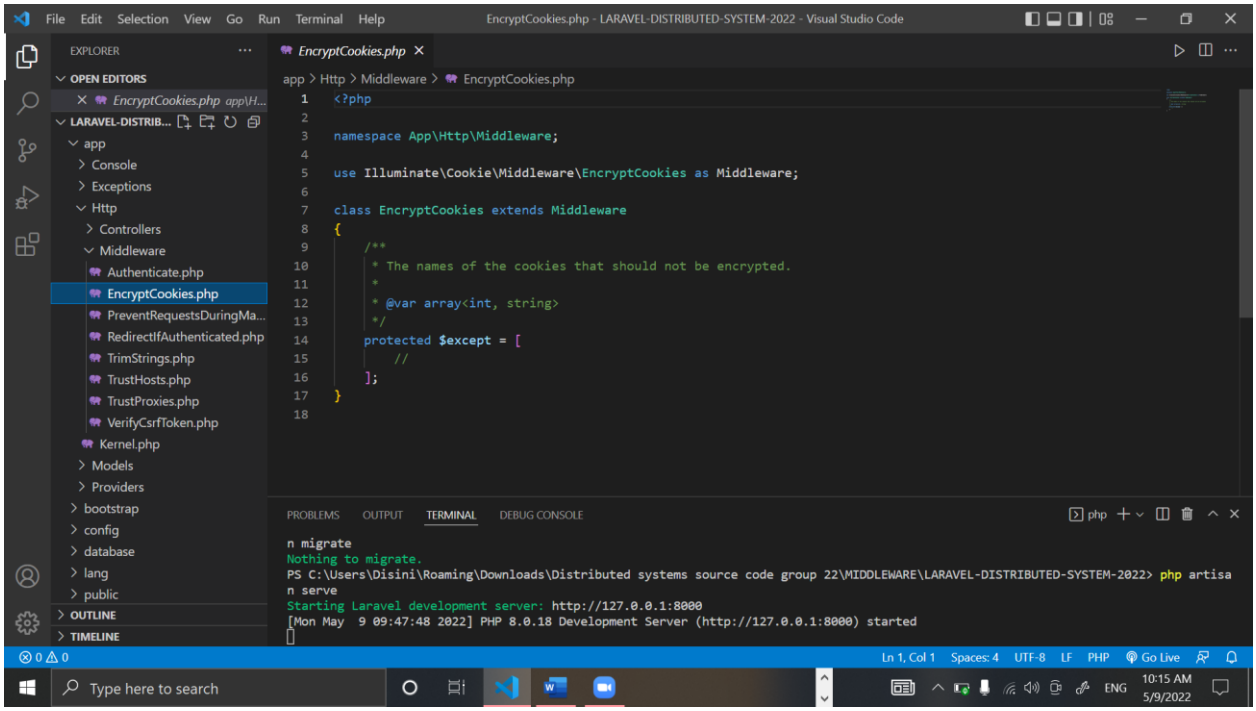


Figure 39: EncryptCookies.php

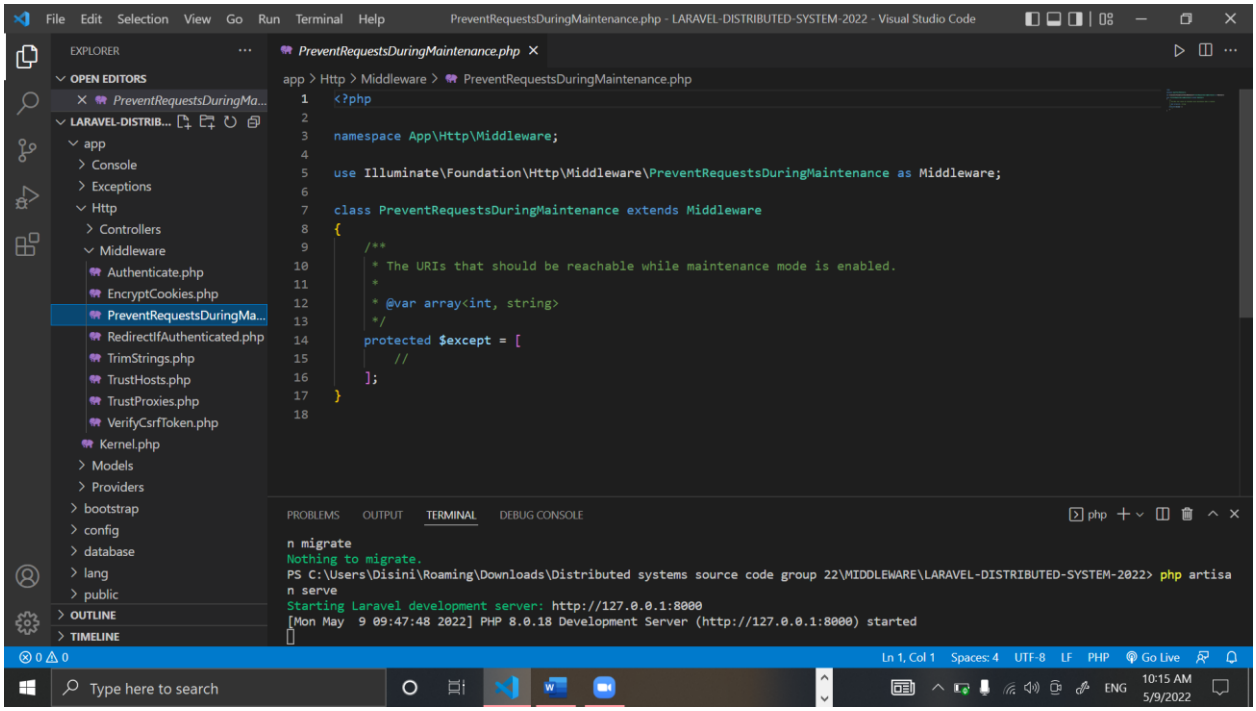


Figure 40: Preventrequestduringmaintenance.php

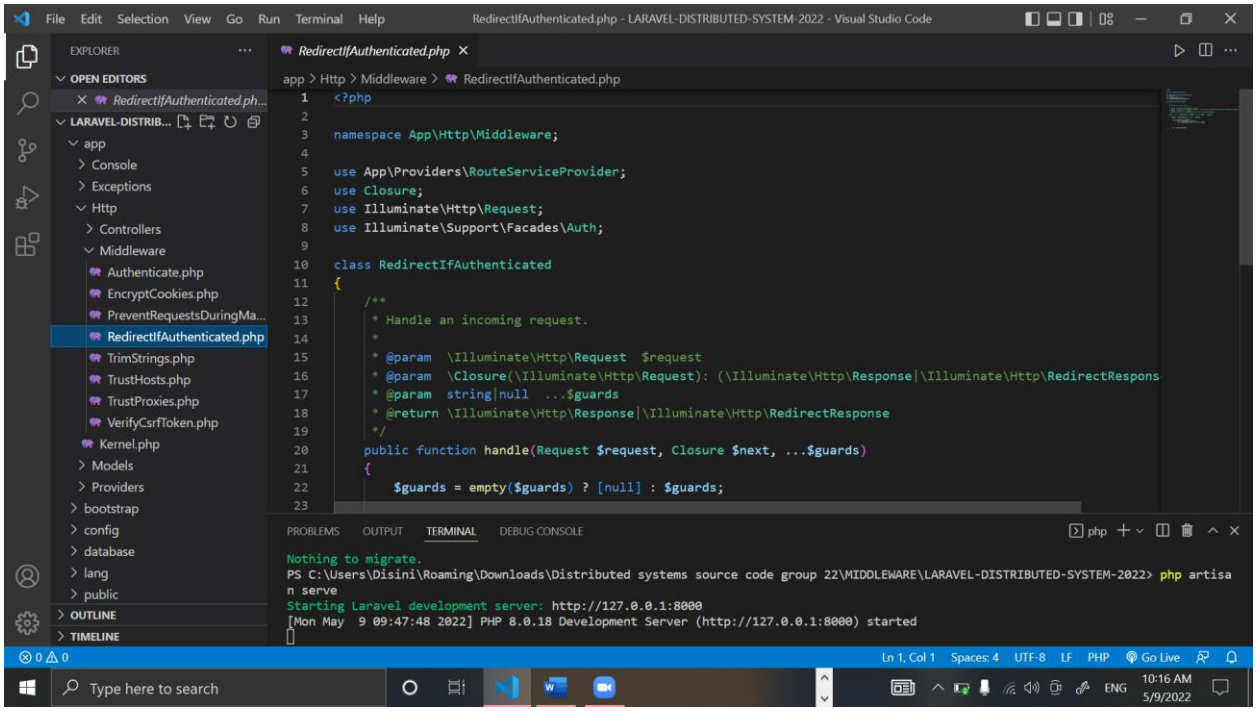


Figure 41: Redirectifauthenticated.php part 1

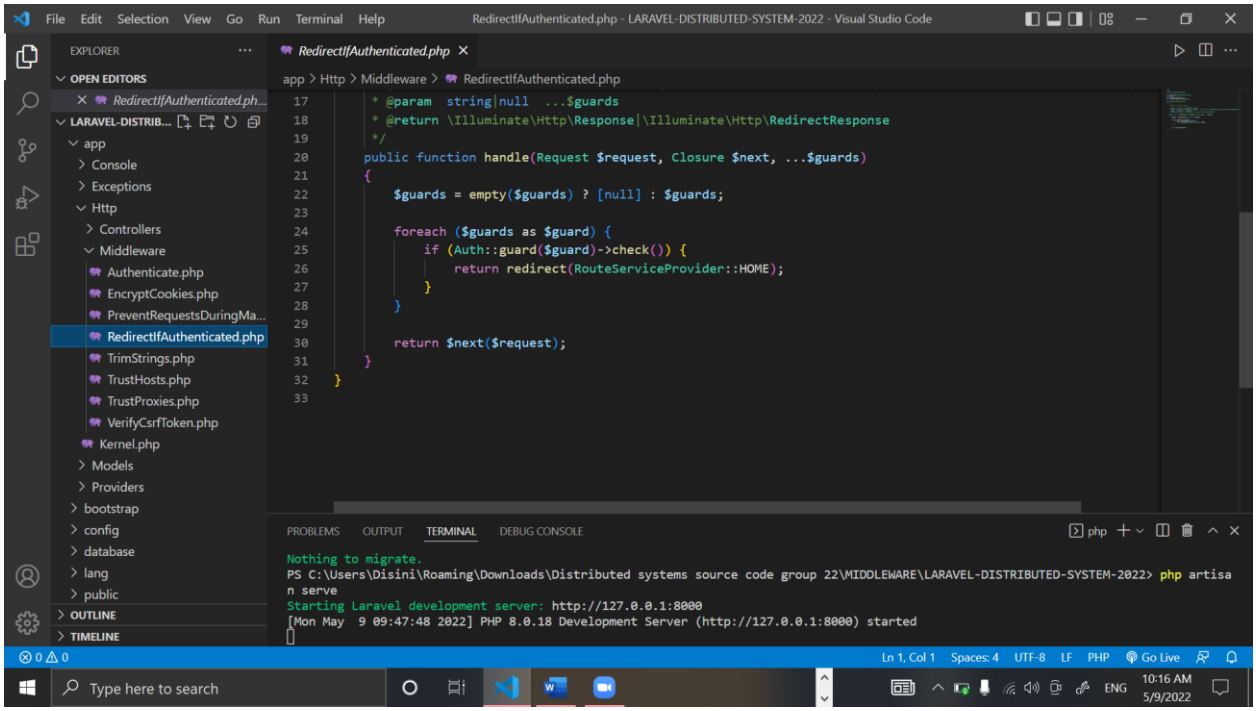


Figure 42: Redirectifauthenticated.php part 2

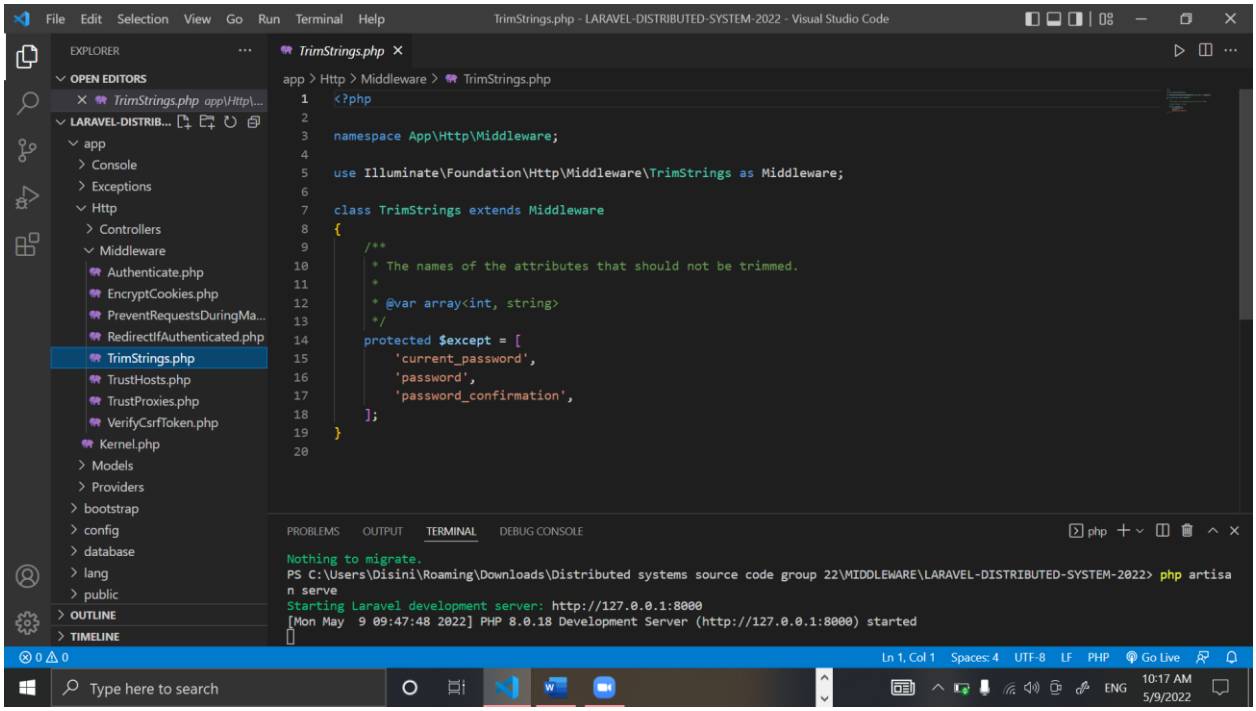


Figure 43: TrimStrings.php

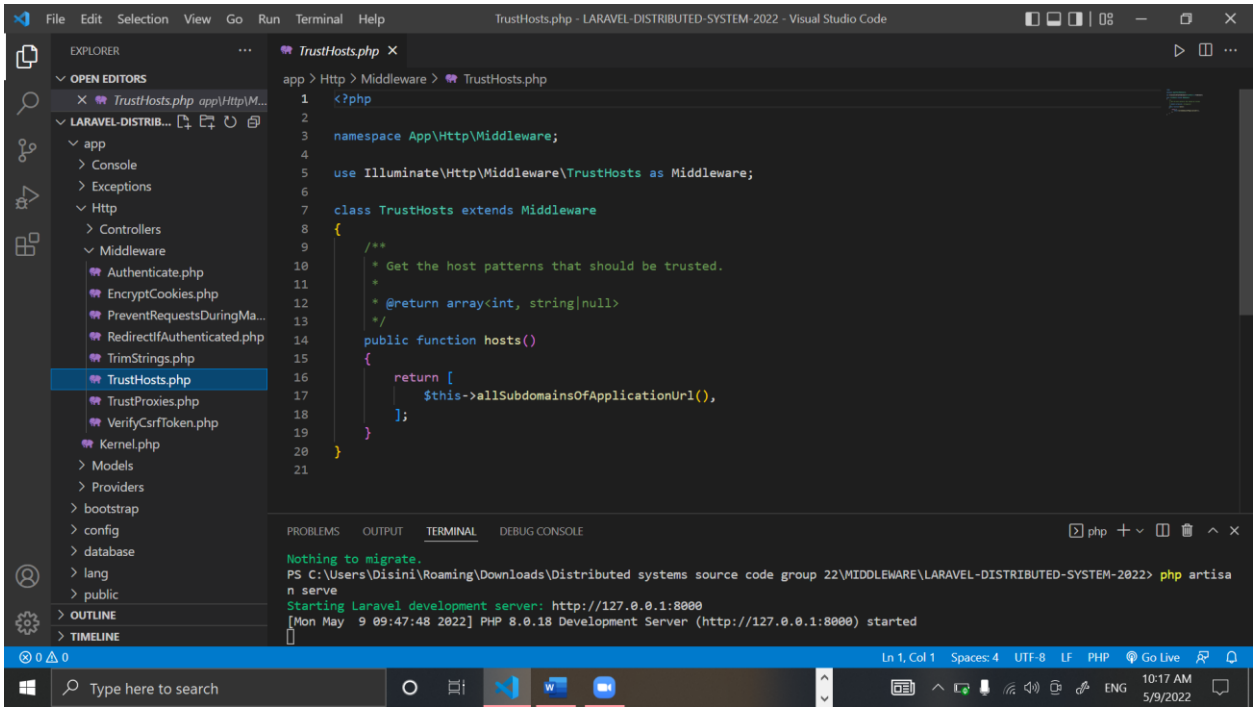


Figure 44: TrustsHosts.php

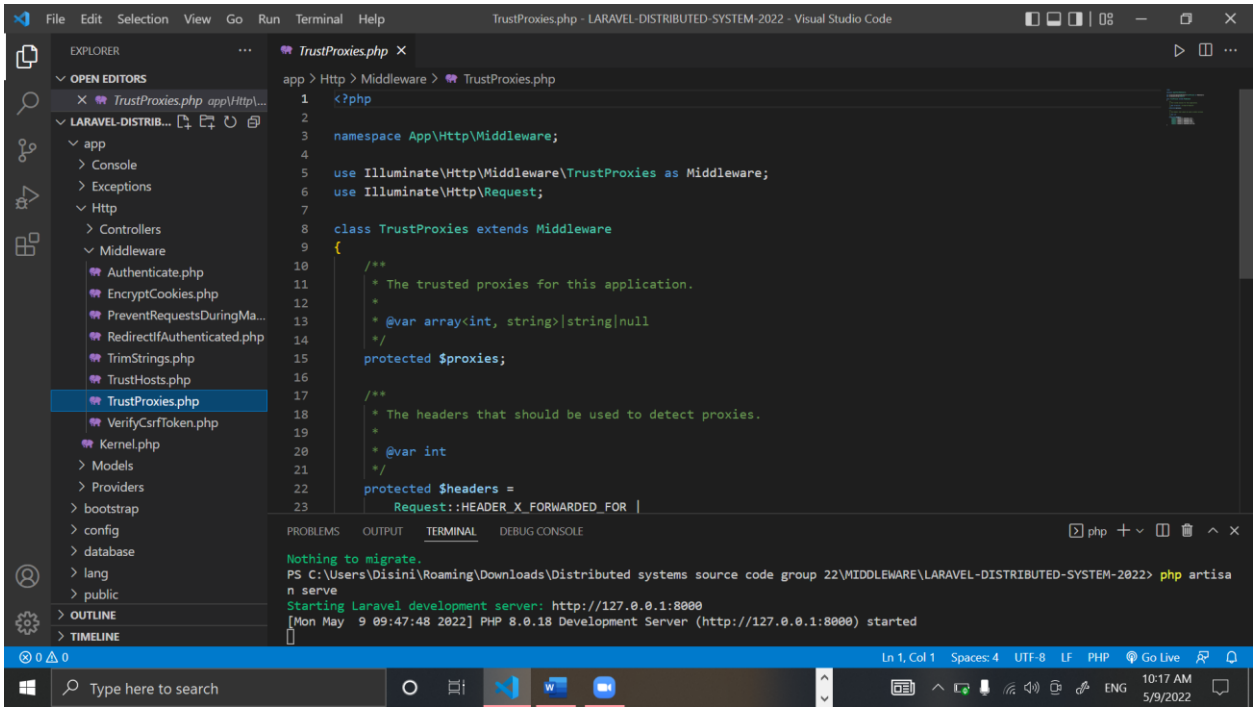


Figure 45: Trustproxies.php part 1

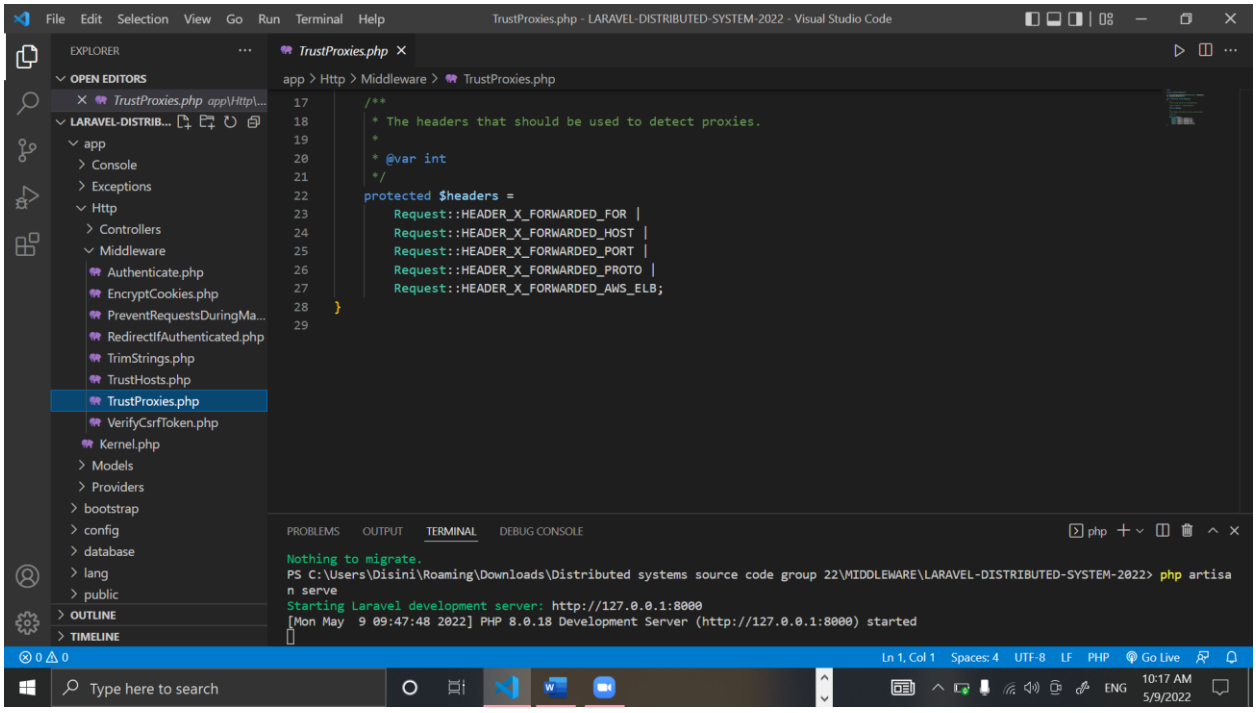


Figure 46: trustproxies.php part 2

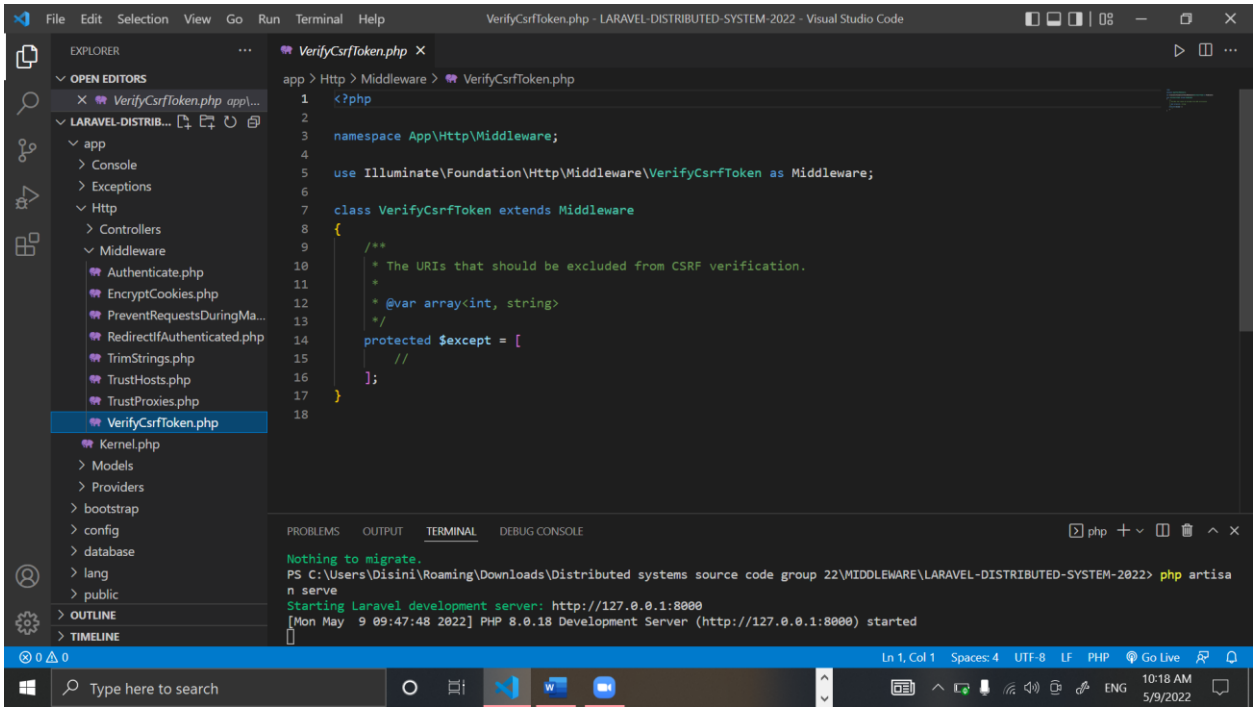


Figure 47: Verifycsrftoken.php

Future enhancement of the system

We as a group would love to upgrade our project if we are to continue this project, this project is to be used by any person who has access to the internet. With this system people can focus and entail in their safety.
We would like to add features were the user can live forecasts and, we would love to add reports from weather stations into the systems, were people can be updated.
We would also like to enhance the project more with checking for weather patterns and collecting data for the betterment of human life.

Risk assessment

Risk	Management Strategy
Risk of Data Loss/ Potential Application Failure	Backup and copy on a storage device.
Overrun on the schedule	The project plan now includes a contingency plan. Highlights and supervisor meetings will be used to track and monitor progress. More days have been added to the estimated completion dates for each step.
Complexity of the project	To adhere to a set of rules that would allow the basic functionality to be implemented and a working product to be produced.
Problems with the technology needed to complete the project	A basic prototype will be constructed for each phase to examine and validate validity. Maintaining a record of each stage.
Other requirements	There will be other projects operating concurrently, and this will receive full attention. A time management strategy would assist me in staying on track.
Lack of information on this topic	• Before starting, do a feasibility study on the topic. • Read books, essays, etc. on the topic.

	• Research on the internet.
Bad Timing	• Plan, then stick to it. • You must determine which project stages or components are most important to you, as well as the short, medium, and long-term impact of each stage/component.
Poor Code Quality	• Review of the code • Coding rules and guidelines that are easy to understand • All code is evaluated. • The Method of Work.

Quality plan

Unit Testing	Separate platforms and code blocks were evaluated in debug mode, bug fixing, and problems were resolved
Integrated Testing	In unit testing as shown with the screen shots of the working application the issues faced with the collaborative workload of the application were fixed and resolved

Table 4: Quality Plan

Summary

To sum everything up, the team has successfully fulfilled the necessary requirements. In comparison to the project proposal, there are some minor changes and deviations in consonance to the utilized programming languages, IDE's and Frameworks. Yet, it is evident that the distributed system has successfully covered the mandatory concepts of building a distributed system.
We have used Laravel to implement our load balancer and middleware.
The overall distributed system is responsive and easy to use. When it comes to fault tolerance, potential network failure is reduced by the addition of one server.
Within the report, we have clearly specified the individual contributions, Architectural Diagrams, middleware and API.

Individual contribution

Member	Index Number	Contribution
J.A. Mujeeb	10707284	I created the dashboard and login in the standalone application created in NetBeans. I also assisted with creating the routes for the API.
P.L. Dilhani	10709402	I created the Load Posts part in the NetBeans application. I also assisted in creating the middleware in the API.
M.D.A. Medhavi	10707278	I created the showing the weather in the web application and the login was done by me. I also assisted in creating the config files.
S.O. Perera	10707315	I created the part where the user can see the posts in the web application and assisted in the creation of config files.
N.S De Alwis	10707160	I created the blog post page in mobile application and dashboard section where users are shown the list of posts related to whether, including the development of the Adapters and Models folders I also assisted in the creation of controllers.

G.M.D.D. Ratnayake	10707351	I created the Signup, Login, and the splash screen in the mobile application, including the development of the Adapters and Models folders. I also assisted in creating the models in the API.

References

https://forge.laravel.com/docs/1.0/servers/load-balancing.html
https://www.w3schools.com/java/java_getstarted.asp
https://laravel.com/docs/9.x/middleware
https://kinsta.com/blog/install-php/
https://medium.com/weblineindia/8-reasons-to-use-java-for-mobile-app-development-87bed4544203#:~:text=The%20primary%20benefit%20of%20using,is%20extensible%2C%20scalable%20and%20adaptable.
Slideshare.net. 2022. Weather Now. [Online] Available at: <https://www.slideshare.net/AmilaWijayarathna/weather-now-75943627> [Accessed 8 May 2022].
https://cms.artio.net/en/what-we-do/software-analysis
https://www.tutorialspoint.com/software_engineering/software_design_basics.htm
https://www.youtube.com/results?search_query=how+to+create+a+distributed+system
https://www.javatpoint.com/software-project-planning
https://www.youtube.com/watch?v=U9P96XQ2688