UNIVERSITY OF LEICESTER

School of Computing and Mathematical Sciences

CO7201 Individual Project

Preliminary Report
AN AUTOMATED SYSTEM FOR LOCAL GPS

[SHREEYA DINESH DESAI]

[sdd13@student.le.ac.uk]

[239038636]

Project Supervisor: [DR. YAKUN JU]
Principal Marker: [DR. STANLEY FUNG]

Word Count: [1400]
[27/02/2025]

DECLARATION
All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: [Shreeya Dinesh Desai]
Date: [27/03/2025]

Contents

1. Overview

This report provides a detailed outline of the progress done between 17th February 2025 to 28th February 2025. The project primarily aims to solve the current problem faced by the healthcare staff and the patients. The objectives aligned with this project include designing user-friendly web application, allowing patients to book appointment without any hassle, doctors/nurses set their availability in advance, manage the prescriptions and deploy the application on the cloud.

Acronyms: WIP[work in progress], TBD[to be done].

2. Progress

Before starting with the development phase, I designed the class diagram, use case diagram and ER Diagrams which are helping to tackle problems faced during project development.

2.1 Essential

- [**Accomplished**]Registration: The patient will be able to register to the webapp using secure login credentials.
  - Frontend (UI): DONE (Appendix: Figure:1, 2, 4)
  - Backend: API developed, and API testing DONE (Appendix: Figure:7 and 8).

- [**WIP**]Availability: The Doctor and Nurses will be allowed to set their availability.
  - Frontend (UI): WIP

- [**Accomplished**]Book appointment: The patients can book the appointment with the available Doctor/Nurse as per their needs.
  - Frontend (UI): DONE

- [**TBD**]Provide prescription: The Doctor will be able to access medical history of patients and provide a digital prescription on the web app.

- [**WIP**]Admin Dashboard: The Admin Dashboard will allow administrators to add, remove healthcare staff and will also allow administrators to book appointments for aged patients.
  - Frontend (UI): WIP

- [**WIP**]Staff Dashboard: The Dashboard will help staff like doctor and nurses to set their availability, view booked appointments, provide prescriptions, view patients medical history and send prescription to pharmacy.
  - Frontend (UI): DONE (Refer Appendix: Figure:5)

- [**WIP**]Patient Dashboard: The Dashboard will show the Doctor/Nurse Availability, book appointment, view prescriptions, upload the prior medical history, previous booked appointments records.
    - Frontend (UI): DONE (Refer Appendix: Figure:3)

2.2 Recommended

- [**TBD**]Deployment on the cloud
- [**TBD**]Articles for minor injuries & awareness: Inorder to tackle mild external injuries, Articles will be provided inorder to take safety measures at home which will reduce some staff workload.
- [**TBD**]View prescription: The patient will be able to view the prescription provided by the Doctor/Nurse online.
- [**TBD**]Buy and Pay prescriptions: For the prescribed medicine the patient can buy and pay for the prescription either online or offline.

2.3 Optional

- [**TBD**]One to one chat: Due shortage of Doctor/Nurse, if in case there's a follow-up required for a specific patient, or a patient requires immediate attention the chat feature can be leverage.
- [**WIP**]Responsive Web Application.
    - The web application is responsive for login, register and all dashboards.
- Video Consultation.

3. Challenges

3.1 Frontend

The challenges faced in the frontend are as follow:

- Designing the Low and High-fidelity prototype and implementing the same was a challenge.
- Styling web pages using CSS was a tedious task, therefore Bootstrap framework was used to design the UI. With the help of bootstrap the time required to style the page was minimal compared to CSS. And to make the web-app responsive, bootstrap was the ideal styling framework.
- Inorder to achieve scalability in near future, transforming the folder structure from simple to organised was a challenge. Another benefit of scalable folder structure was clean code and easier to understand by other developers.
- With growing features mapping React components to their appropriate pages was a challenging task.
- Designing the TopNavBar and LeftNavBar (refer to Appendix: Figure:6) was the most challenging part. The challenge faced was to develop logic for the

navbar to be functional even on mobile view. Several online articles were referred to develop the business logic.

Cloud Deployment:

Deployment of the frontend of the web app was done inorder to understand the deployment process. The cloud which I have opted for is Azure, which again was a challenging decision as there are several cloud providers in the market. After development of a minimal ready-to-deploy web app, understanding and learning the ways to deploy frontend was a crucial part. After browsing through various articles, the decision for docker deployment is considered.

There was a need to learn docker and azure deployment services together, these tasks required a lot of time and study. Finally, after few trials and errors, I was able to deploy the application and get the outcome. (Refer to Appendix: Figure 9 & Figure 10)

Backend:

Backend Development is new to me when compared to other two tasks. In the beginning, understanding the problem statement and designing data models was a serious task. For the development of the web-app, I chose to go ahead with Database-first approach, as I had already read about it and was familiar with the process.

Following there was a need to opt for backend language, initially for backend development JavaScript seemed new and easy to learn. However, in my previous academic projects I have developed backend logics using Python and Flask, so I decided to stick to it.
SQL SERVER was chosen as the ideal database for our project, the initial connection between Flask API and SQL SERVER was an everlasting process. Further down the line, once the connection was established developing API was not a hard task. As this has been said, there are some API which require complex logic and problem-solving skills like staff login.

4. Updated Time Plan

Week 1: 10/02/2025 - 16/02/2025

- ***Milestone Achieved: Planning and Requirement Analysis (Stage 1).***

Supervision meeting done.
ER-Diagram, Use-Case Diagram done.
Project Brief done.

Week 2: 17/02/2025 - 23/02/2025

- ***Milestone Achieved: Defining Requirements (Stage 2) & Design (Stage 3).***

Iterated the Er-Diagram, Use-case diagram, Class Diagram.
Preliminary Report draft.
Wireframes for the webpage.

Week 3: 24/02/2025 - 02/03/2025

- ***Milestone Achieved: Start of Development (stage 4)***

Supervision meeting done.
Preliminary Report done.
High-Fidelity Prototypes.
Frontend: Login/Register page done.

Week 4: 03/03/2025 - 09/03/2025
Added Bootstrap: Login & Register page.
Dashboard Prototype.
TopNavbar and LeftNavBar Completed.

Week 5: 10/03/2025 - 16/03/2025
Deployed Frontend.
SQL Scripts.
Backend: File Setup.

Week 6: 17/03/2025 - 23/03/2025
Edited Frontend Structure.
API testing.
Database: Patient table.
Backend: Patient Login and Register API.
Frontend: Completed Staff login and Patient Login.

NOTE: The Project is delayed by 1 and a half week.
Below mentioned is the Updated Time Plan.

Week 7: 24/03/2025 - 30/03/2025
Frontend: Patient and Staff Dashboard.
The backend logic for book appointments and set availability will be completed.
The frontend book appointments, set availability, add patients and doctor will be completed, and integration of frontend and backend will be performed.
Therefore, the interim report will be submitted (28th March 2025).

Week 8: 31/03/2025 - 06/04/2025
The principal marker interview is scheduled on 31st March 2025.
Previous backlogs will be covered which will include the manage prescriptions and implementation of one-to-one chat will be initiated.
The implementation of article for minor injuries will be done.

Week 9: 07/04/2025 - 13/04/2025

- ***Milestone Achieved: Essential Requirement Achieved along with few of the Recommended and Optional feature.***

- ***Milestone Achieved: Testing (Stage 5).***

The supervisor (in-person) meeting will be held where the working application will be shown. The web application will be tested and if in case any backlogs are remaining it will be completed during this week. If time persist, the cloud architecture will be designed which will help to deploy the application.

Week 10 & Week 11: 14/04/2025 - 27/04/2025

- ***Milestone Achieved: Deployment and Maintenance (Stage 6).***

The supervisor (in-person) meeting will be held and during this two week the application will be successfully deployed to the cloud and the final report template will be drafted and submitted (25th April 2025).

Week 12: 28/04/2025 - 04/05/2025

The group supervisor meeting will be held, and the previous backlogs will be completed along with start of the final report.

Week 13: 05/05/2025 - 11/05/2025

The supervisor (in-person) meeting will be held. During this week, continuation of final report and testing of the deployed application will be performed.

Week 14: 12/05/2025 - 18/05/2025

In this week, the final report and code will be submitted (16th May 2025).

Week 15: 19/05/2025 - 23/05/2025

In this week, the final viva will be conducted.

5. Appendix

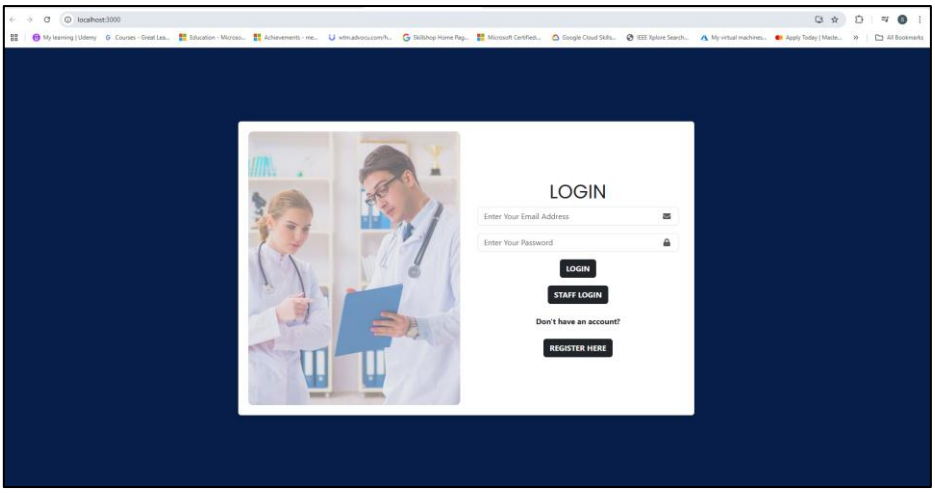

Figure 1: **Login Page.**

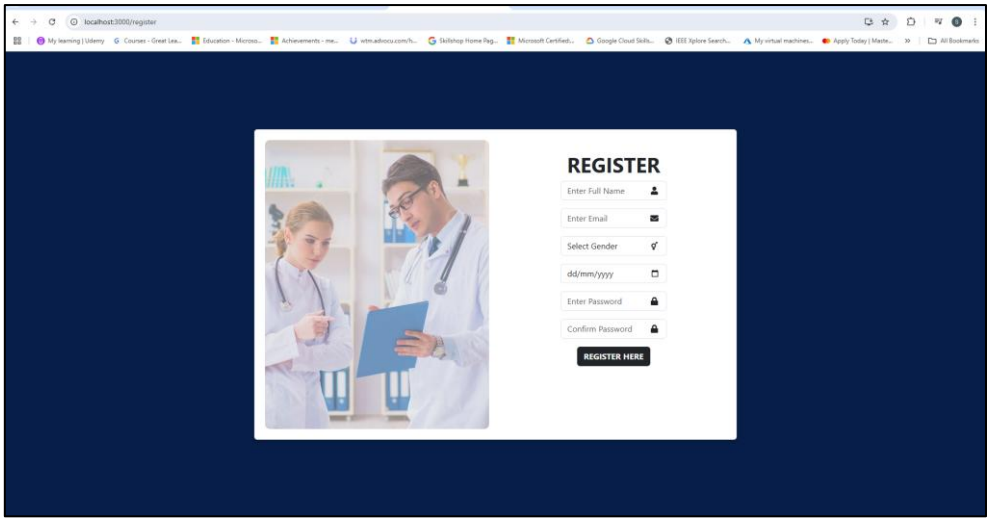

***Figure 2:*Patient Register From**

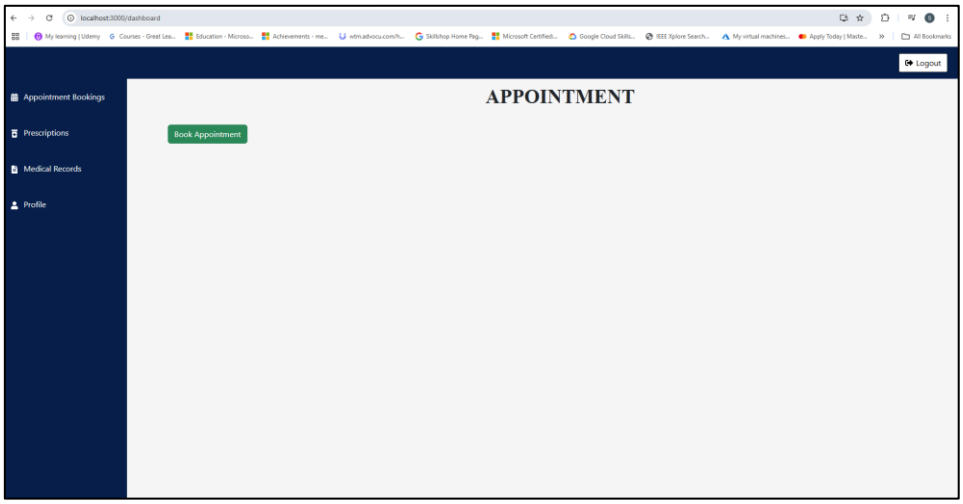

***Figure 3:*Patient Dashboard**

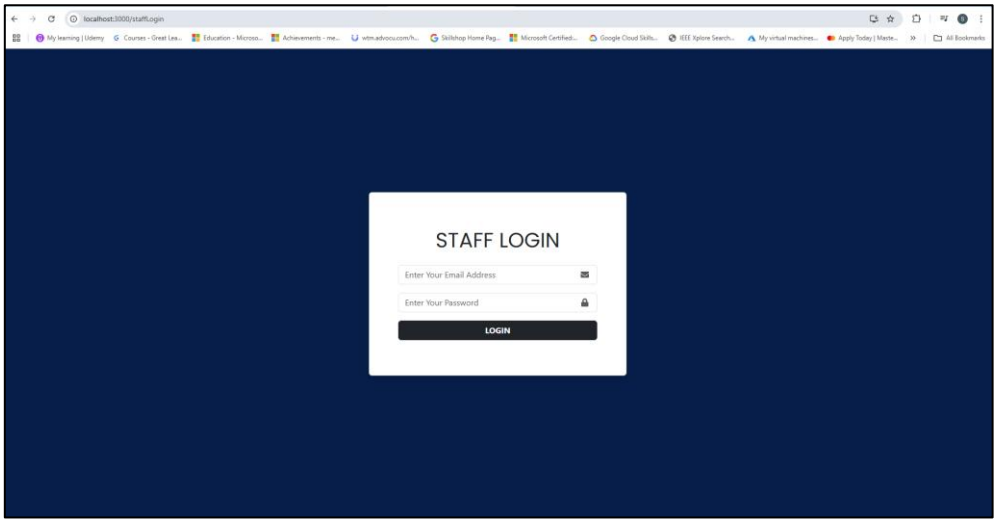

***Figure 4:*Staff Login**

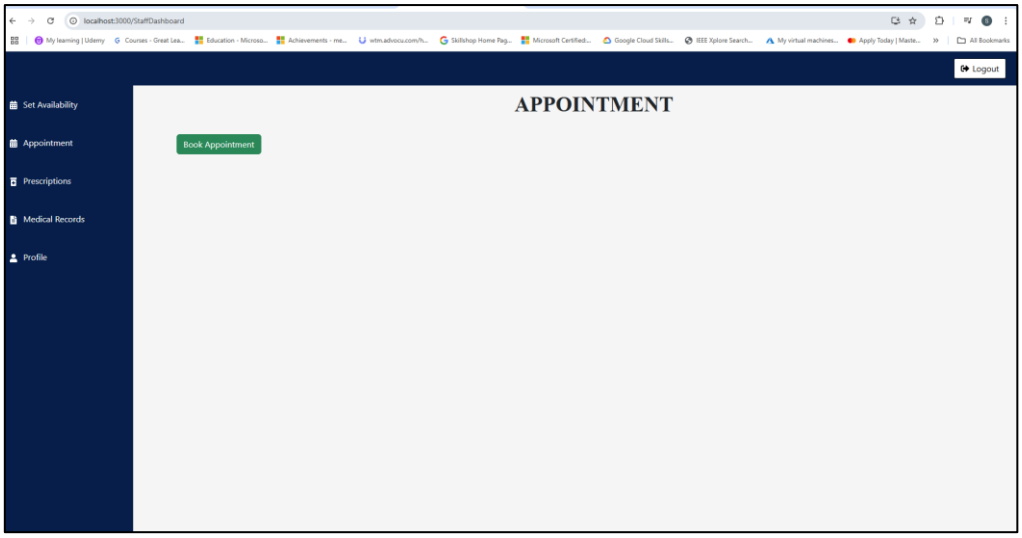

Figure 5: **Staff Dashboard**

```
const Dashboard = () => {
  const [isSidebarOpen, setIsSidebarOpen] = useState(false);
  const [appointmentData, setAppointmentData] = useState(null);

  const toggleSidebar = () => {
    setIsSidebarOpen(!isSidebarOpen);
  };

  const closeSidebar = () => {
    setIsSidebarOpen(false);
  };
```

```
    <TopNavbar onToggleSidebar={toggleSidebar} />
    <LeftNavbar
      isOpen={isSidebarOpen}
      onClose={closeSidebar}
      links={navbarLinks}
    />
```

Figure 6: **TopNavBar & LeftNavBar Code Snippet.**

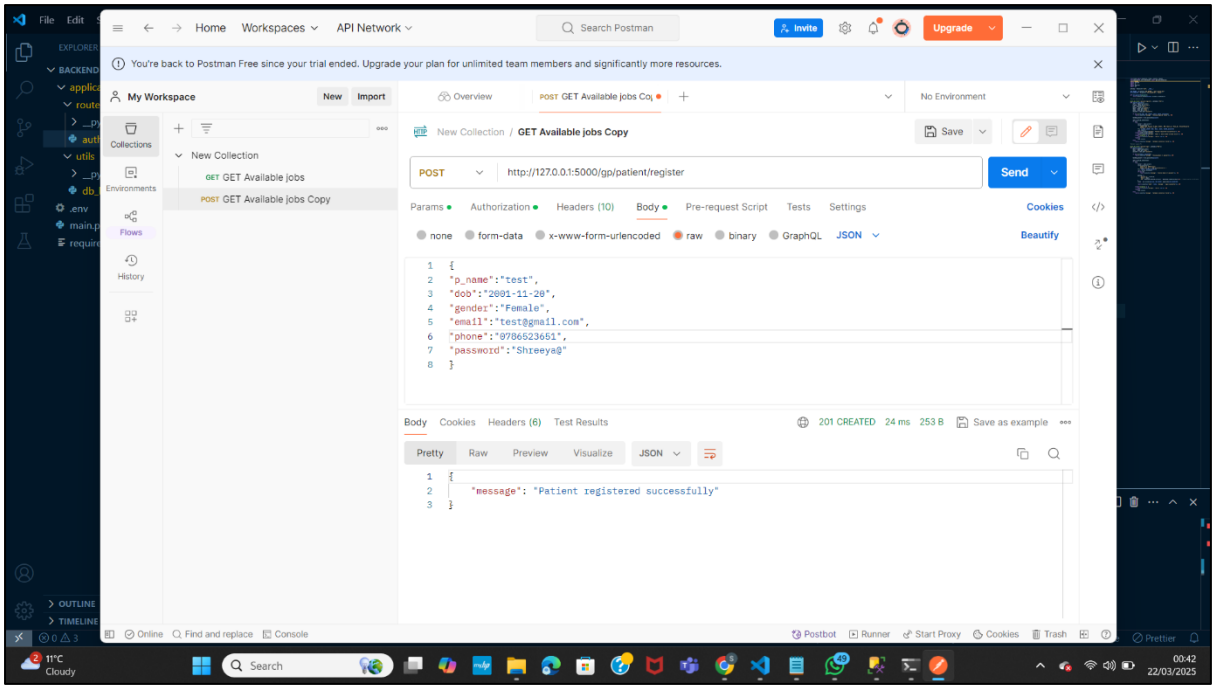

Figure 7: **API Testing**

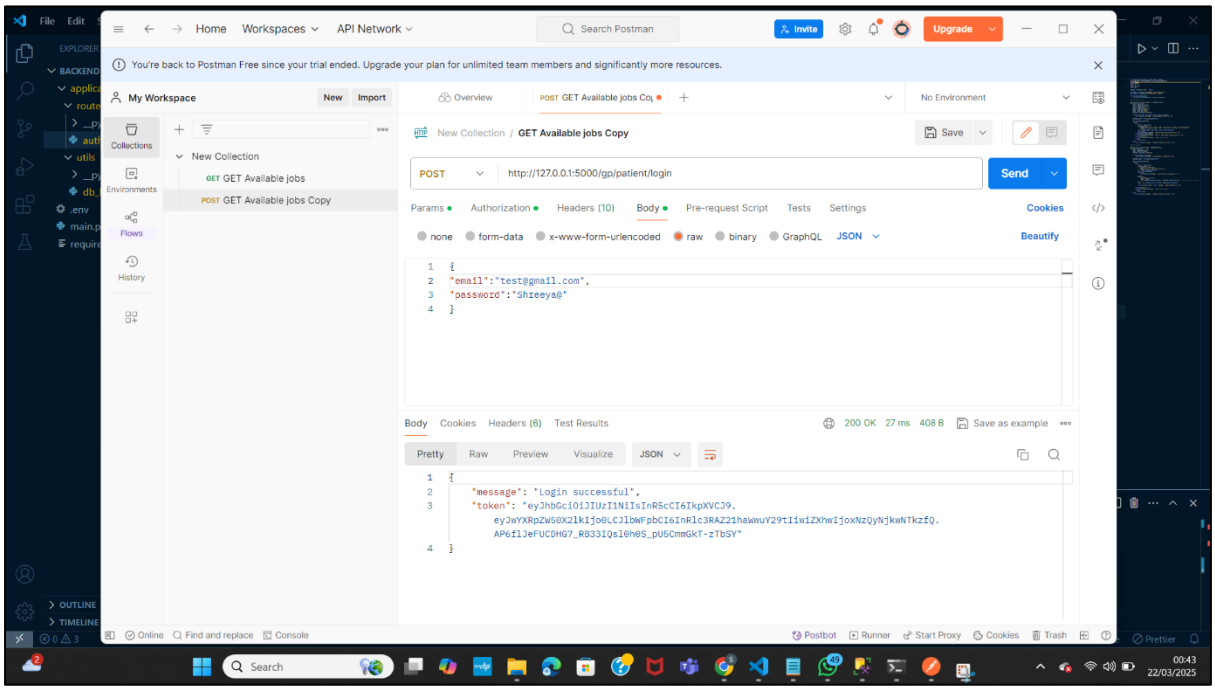

Figure 8: **API Testing**

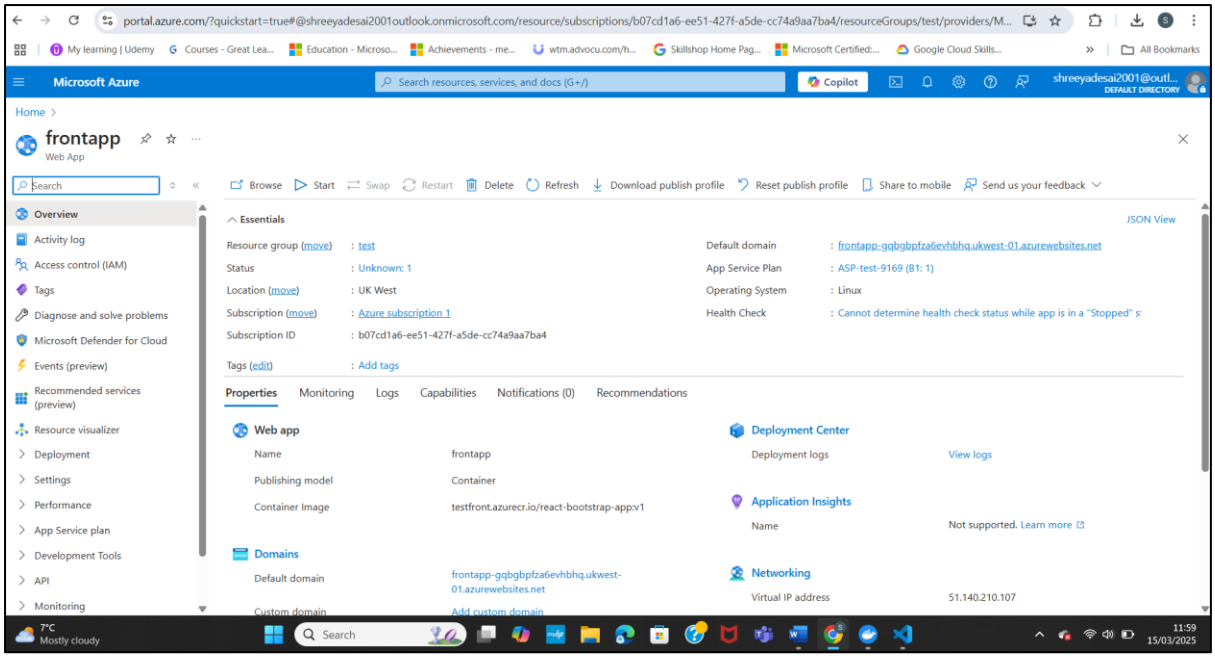

Figure 9: **Azure Dashboard**

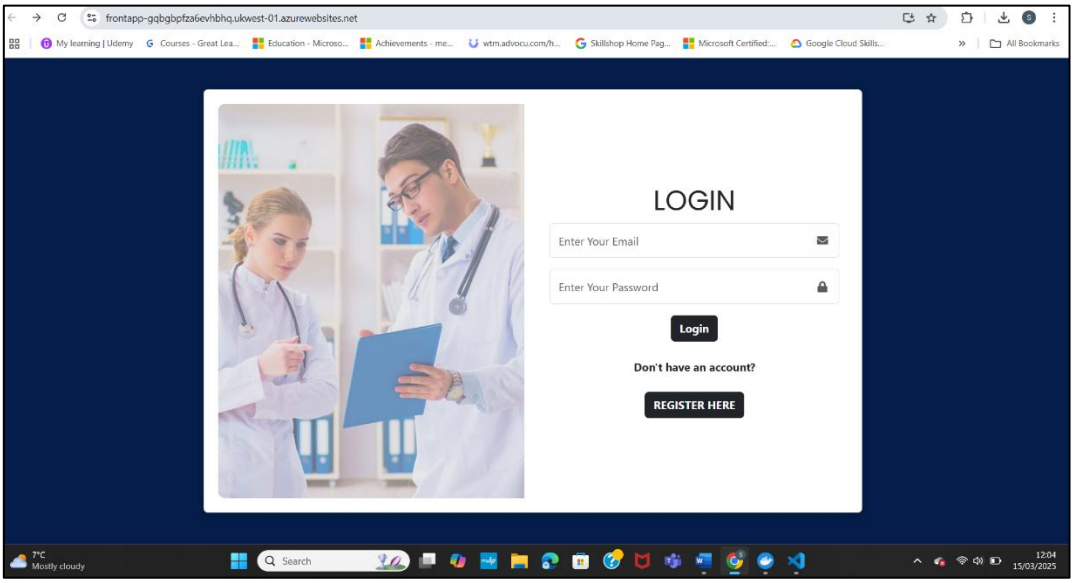

*Figure 10:***Deployment Result**