MedVoyage

Group Name: Squad8

Team Members: Emmie Abels, Kalyan Cheerla, Vidhi Bhatt, Yasmeen Haleem, Bhavani Rachakatla, Demir Altay, Manushree Buyya, Pravallika Bollavaram

Project Description

Have you ever felt low and absolutely could not find the motivation to pick up the phone and go through the long and tedious process of contacting your doctor? Even if you pick up the phone, you are often left on voicemail. Not only does this waste your time and energy, it also does not provide efficient or practical results. Now you are wondering what if you could easily and quickly book a doctor's appointment that assists you with your concerns?

This is possible with MedVoyage! Through MedVoyage, our mission is to serve patients and their needs and provide quality customer service in a completely accessible, convenient, and exceptional way. MedVoyage provides an easy sign-up process with an email and a strong password. After this, you will be asked to enter your personal information, including your name, address, email address, phone number, insurance information, birth date, allergies, medical history, etc. Once you have filled out this section, the web app presents the user with multiple questions regarding your preferences for a doctor. This includes whether you prefer your doctor to be male, female, or non-binary, and the system will do its best to connect you with your selected preferences. Other questions include the type of specialty doctor you need. For example, if you have skin concerns, a drop-down arrow in the section encompassing a doctor's specialty will permit you to pick a dermatologist. If you want to see a primary care physician for a routine yearly checkup, you can select "Primary Care Physician" in the list of options.

Added preference-related questions include how urgent your concern is and how urgent your appointment is to be; for example, if a patient needs to visit the doctor in two days, they can select the possibility of "in 48 hours (about 2 days)" in the "How Urgent" section of the patient preferences portal. Through this, a list of the appropriate doctors that match the patient's time concerns and needs will be presented to the patient. Furthermore, as the patient inserts their address into the MedVoyage personal information portal, the user will be directed to the doctors in the nearby radius that they have chosen. For example, a patient can pick a radius of 25 miles, and all the nearby doctors that meet this and other patient requirements will be available to select and book an appointment with. The user is also asked to insert their insurance information, which is vital, and with the given insurance information, the user will be directed to doctors who accept their insurance company. The patient's medical history is something for the doctors to include in their chart, which reduces patient waiting time for the appointment in the waiting room filling out patient information and history paperwork. Overall, we hope to match a patient to the doctor that meets most of their needs at their convenient time so they can fit in with their busy schedules. We hope to provide a fulfilling client-doctor connection where the doctors or their teams can access patient information easily and the patient is left satisfied in the process of finding and booking a doctor. This is described in detail in the project concept section below.

Project Concept

The system will also have various features for doctors to use to help them manage their time and the patients. Our booking system will be designed for use by both patients and doctors. When patients use the system, they will be able to book an appointment with any doctor in the system. The system will show the patients what appointments are available for each doctor and will be accessible. Patients will also have a personal profile that holds the patient's personal information and insurance information store so that the patient does not have to enter the information every time they want to make an appointment. The system will also have various features for doctor can use to help them manage their time and the patients.

With our system, doctors will be able to set their schedules. They can block off times and days for certain types of appointments and decide how long each is so that both patients and doctors are efficiently using their time. Doctors will also be able to see their entire schedule for any day, week, or month at any time and will be able to easily pull up the patient information for all their patients to help them prepare for a patient's appointment.

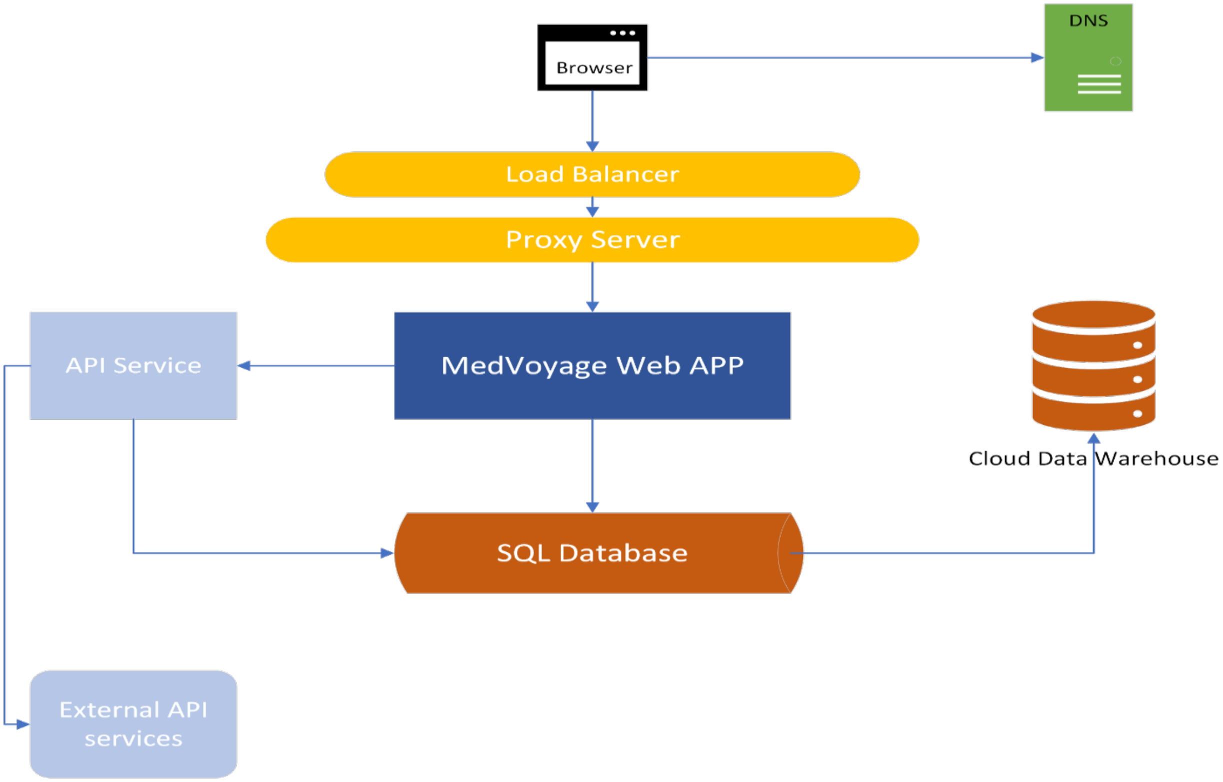

Fig 1: High-Level Design

Above Figure 1, has details regarding the High-Level Design of our project which also explains details on how both the clients and doctors would use the platform from a technical standpoint. Here, the whole MedVoyage webapp is placed behind a proxy server like Nginx or Apache2 and a load balancer. The request flows through the load balancer and proxy server and then to the main webapp. For handling any DB details, the main webapp will be integrated with a SQL database. Also, for the main webapp we are trying to go with Django web framework which would be able to handle login requests and other maintenance details for the webapp. We also had thoughts on integrating external API services for handling payments, email, and message notifications for the webapp. Additionally, there might be other proprietary API services that can be designed for specific use cases like providing DB data access to perform data analysis for useful ML algorithms that can be achieved using Flask web framework.

Languages and Technologies

We are planning to use HTML, CSS, Bootstrap, and some JavaScript as frontend. Django or Flask as our full-fledged web frameworks through templates for its ease of use, mobile first development, and many other libraries that can be used to enhance our web application to the best possible version of itself.

For the hosting we are planning to use GCP (Google Cloud Platform) and for the backend programming language we will use Python and Django or Flask which would handle the REST web requests. For the Database we are going with MySQL and exploring the free Firestore database provided through Firestore by Google Inc.

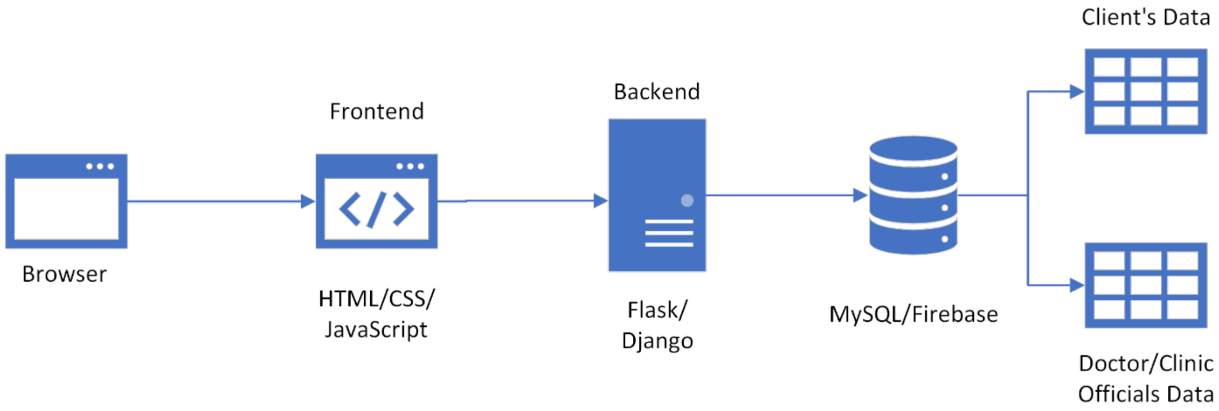

Fig 2: Details about front-end, back-end, database

Above Figure 2 clearly shows where our different frontend, back-end and database components in the project would go. As explained in the earlier sections our project handles data from both Doctors/Clinic officials and the Clinic users/clients. Hence, we would like to segregate these into different databases in MySQL or tables in the database as needed.

Below we will go more into depth about what GCP (Google Cloud Platform), and Python are and their benefits for this project.

What is GCP?

- Google Cloud Platform is a comprehensive cloud computing platform provided by google that offers a wide array of cloud services, including computing, storage, databases, machine learning and more. One of its primary advantages is its global network infrastructure, which includes many datacenters strategically placed around the globe. The infrastructure ensures low latency access to resources and high availability of services. GCP also offers a wide array of services including Google Kubernetes Engine for cluster organization, Big Query for data analytics, and AI/ML services like TensorFlow and AI Platform. Below we will go into a bit more detail about how some of these services and features will allow us to take this project to the next level.

Scalability and Performance

- GCP's scalability features are particularly advantageous for a doctor's booking webapp. Healthcare appointments can experience varying levels of demand, such as peak times for flu shots or routine check-ups. With GCP, the app can automatically scale its resources up during peak times and scale down seamlessly during quiet times, saving us money eventually all while ensuring our users get the best experience possible, regardless of how many users are active.

Machine Learning for Optimization (stretch goal but possible with GCP)

- GCP's machine learning capabilities can be harnessed to optimize various parts of the booking process. For example, ML algorithms can predict appointment demand based on historical data helping doctors and clinics staff appropriately. It could also suggest suitable time slots based on patient preferences and availability.

Global Reach

If our webapp were to serve patients across different geographical regions, GCP's global network infrastructure would ensure that a patient in Denton Texas would get the same experience as a patient in Istanbul Turkey for example. This would in turn enhance user satisfaction while ensuring appointment scheduling.

In summary Google Cloud Platform offers significant advantages for a doctor's booking web app like ours, including dynamic scalability, robust security and compliance features, advanced machine learning capabilities, and a global network infrastructure. These features will help ensure a smooth operation of the web app while maintaining data security and legal compliance in the healthcare industry.

We are planning to use either Flask or Django or both in our backend for the following reason below:

Flask:

1. Lightweight and Flexible: Flask is a micro-framework, making it lightweight and flexible. For a doctor's booking web app like ours, this flexibility will allow us to choose and integrate specific components tailored to our project's requirements. We can add extra extensions as needed, keeping the application minimalistic and efficient.
2. Modular Design: Flask encourages a modular approach to building web applications. This means we can break down the app into smaller, more manageable components, which is particularly useful in a healthcare app like ours which we can see becoming overly complex very quickly.
3. Ease of Learning: Flask's simplicity and minimalistic design makes it easy to learn, which will be advantageous since a few people on our team have no prior knowledge of the framework.

Django:

1. High-Level Abstractions: Django is a high-level web framework that provides a robust set of built-in features and abstractions. This can significantly accelerate development for our app, as many common web development tasks like user authentication, database management, and form handling are handled out of the box.
2. Security and Built-in Authentication: Security is of utmost importance in healthcare applications. Django has built-in security features such as protection against common web vulnerabilities like SQL injection, cross-site scripting (XSS). It also provides a robust authentication system, which is crucial for ensuring that patient and doctors information is kept confidential.
3. Admin Interface: Django's admin interface provides a ready-made solution for managing the backend of our application. This will be extremely useful for the healthcare administrators and staff who need to manage appointments, patient records, and doctor's profiles efficiently.

In summary using both Flask and Django in combination can be a powerful approach for our app. Flask's flexibility and modularity allow us to design the API services in our application to meet specific requirements, while Django's high-level abstractions, security features, and admin interface can accelerate development and ensure robustness in managing sensitive healthcare data for the main webapp.

A Focus on Mobile Accessibility

For our project, we have decided to create a webapp that is mobile friendly. We felt that MedVoyage should be easily accessible from any device so patients are able to book an appointment wherever they are. The webapp will focus on being mobile friendly since many people often use their phones or tablets to book appointments. When a person is sick, the last thing they want to do is hassle with making an appointment on a computer so making the webapp mobile friendly allows for a better user experience for patients.

Many clinics often use tablets with doctors and patients due to how portable they are and because they are easy to clean. A mobile friendly web app ensures that any clinic and doctor can see their patient's information and schedule throughout the day with ease.

Development Environments

As a development environment, we will use Visual Studio Code, PyCharm, as we know VS code editor is a free open-source editor designed by Microsoft. It is a light weighted and extensible design that makes it a popular choice among developers to support various programming languages. PyCharm is an integrated development environment (IDE) specifically designed for Python development. It is developed by JetBrains and comes in two editions, Community (free) and Professional (paid). For our project, we will use a free community version which provides support for web applications and python centric.

Project Timeline

In the first phase, "Project Planning and Proposal," the groundwork for the project is established. This includes setting up essential tools such as GitHub and Trello for collaboration, preparing a comprehensive project proposal, creating a Risk Management Plan to address potential risks, and crafting a presentation and video to communicate the project's essence.

The second phase, "System Specifications and Requirements," defines the project's specifications and requirements. This entails thorough requirement analysis, drafting system specifications, and refining them to ensure clarity and completeness.

Moving on to the third phase, "Project Launch and Execution," the actual project development begins. This phase involves implementing front-end and backend features for clients and healthcare professionals, alongside rigorous testing, and draft report generation to monitor progress.

In the fourth phase, "Project Execution," critical functionalities such as appointment scheduling and tracking are implemented, and the web application is deployed. Further testing and draft reports are conducted to maintain quality control.

Finally, in the fifth phase, "Project Testing & Deployment," the project undergoes extensive system and usability testing. It culminates with the deployment of the web application, along with comprehensive documentation, project reports, and a final presentation to showcase the completed project. These phases and subtopics together ensure a systematic and well-structured approach to project development.

In addition to outlining the project's tasks and subtopics, we have created a Gantt chart to represent the project timeline and milestones visually. The Gantt chart serves as a valuable tool for tracking the progress of each phase and ensuring that tasks are completed in the correct sequence. It helps us identify critical dependencies and allocate resources efficiently to meet our project objectives within the defined timeline. The Gantt chart provides a clear overview of the project's schedule, making it easier for us to understand our project's progress and key milestones.

The following is our project timeline Gantt chart:

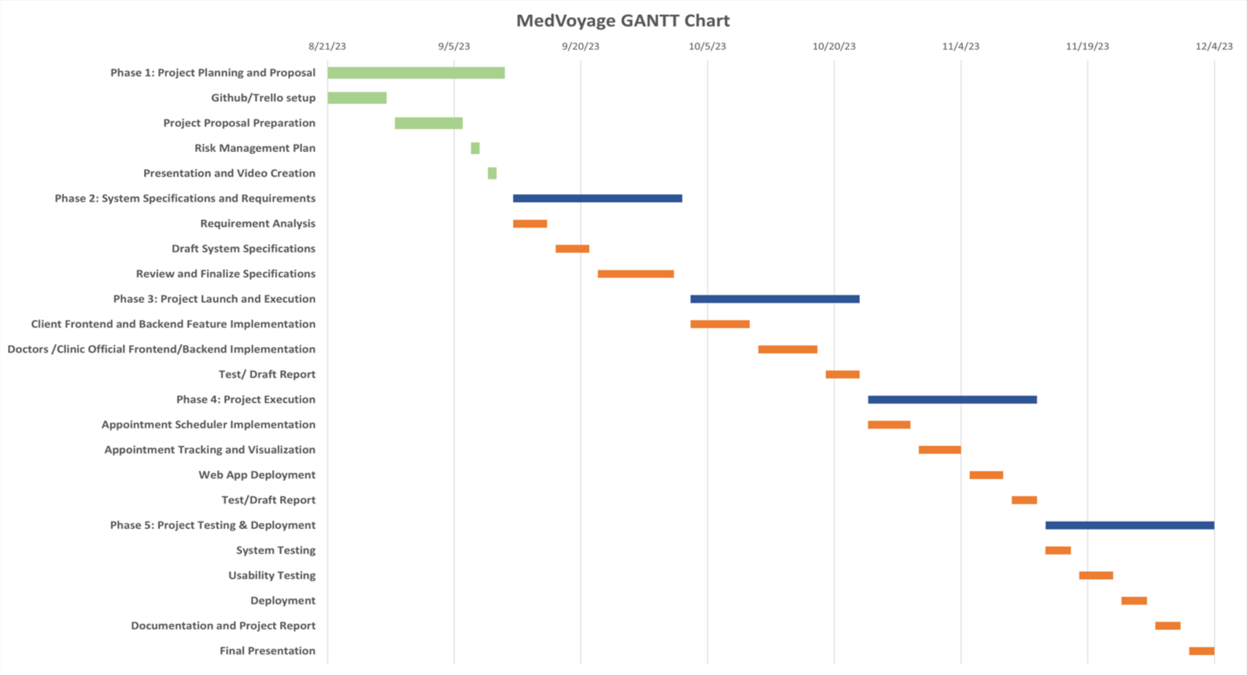

Fig 3: Gantt Chart

🟩 Completed Tasks
🟦 Major Deliverables
🟧 Sub Tasks

Risk Management

Top 3 Risks

Developing a web-based online clinic booking application comes with various risks that should be carefully considered during the project planning and execution phases.

The top three risks for the med voyage project are:

1. Privacy:

As patients will be booking appointments with the doctors on the MedVoyage app, patient's personal and health information will be uploaded on the webapp. As per HIPAA rule patient's information is protected by certain guidelines. Care must be taken to prevent unauthorized access to patient data and data breaches.

2. Resource Access Issues:

There can be certain risks associated with cloud resource availability, scalability issues, issues related to server downtime, maintenance or infrastructure failures, issues related to integration with third party services, software bugs.

3. Data Handling and Password:

The data of the customers and the clinic and doctor information should be secured, to prevent accidental data loss or corruption due to human error, hardware failure or software bugs which can lead to data unavailability or integrity issues. Apart from the above-mentioned risks, payment handling and secure payments and intense competition in the health care domain are also possible.

Risk Monitoring

For privacy risks, a plan must be made to keep updated with changes in health care regulations and compliances. By identifying the different resource shortages, we can monitor resource access issues. We plan to monitor privacy or data handling risks by first identifying what type of privacy or data handing issues our program is having once we have the program running, and then by accessing them with the approaches described below and in a way that our contingency plan and strategies are changing over time. One can make sure any type of breach is being immediately reported. Apply a type of design that ensures security and privacy, can also establish a generic terms and conditions statement, making sure the users are aware of how their data is being utilized, and if some users choose to not share that information, they will have an option not to. We plan to create an analysis documentation and report risks related to resource shortages, data handling, and privacy concerns, then provide documentation based on how often they occur in our program and how serious they are. Given these risks are our top three serious risks, we created the below plans to make sure they are taken care of; however, if we see something unique, we plan to redevelop the contingency plans below based on our analysis of how often and how detrimental these risks are. Through this analyzation of our risks, we can monitor risk frequency and pay attention to what other than common identifiable risks are possible in our program.

For data handling and password risks, the security measures, access logs, and detection systems can be used to identify any breach, and continuously accessing the compliance with data protection regulations and by staying updated on real-time and latest threat and implementing data backups and monitoring back log for errors.

Risk Reevaluation

1. Protection of Patient Information (HIPAA Compliance): Continuously monitor and assess data security measures throughout the semester to maintain HIPAA compliance. Periodically review access controls and encryption protocols to adapt to changing threats and regulations.

2. Resource Access Issues (Cloud and Server-Related): Maintain constant monitoring of cloud resources and servers. Be prepared to adjust to changing demands and downtime. Regularly assess resource utilization and redundancy measures as needed for seamless access.

3. Data Handling and Password Security: Conduct data backups and integrity checks to prevent data loss or corruption. Monitor password security practices and update policies as required. Re-evaluate data protection measures and alignment with best practices to ensure data availability and integrity.

Contingency Plans

The contingency plan regarding the privacy of the patient's information can be handled by cryptography and encryption and securing the patient's details in accordance with the HIPAA rules.

To ensure that all the resources like databases and cloud services are running at any given time, and to prevent issues related to resource unavailability, we would like to implement GCP failover.

To ensure safety for data handling and password protection, certain methods like Duo multi factor authentication and encryption of data will be performed.

To prevent payment-related issues dealing with payment cards PCI-DSS compliance standards and rules and regulations will be met.

Team Member Roles

All team members will be responsible for coding during this project. In addition, the following is the roles and responsibilities of each team member:

Emmie Abels
Throughout the project, Emmie will be working with Kalyan as part of the Project Management Team. She will also be the front-end Requirements and Design Lead for the project. Emmie has previous experience with front-end web development so our team felt that these would be the best roles for her to take during this project. Due to her experience with usability testing and QA/QC she will also be a leading member of the Testing Team.

Kalyan Cheerla
Kalyan will be working as the Head of Project Management Team driving the conversations during the meetings and monitoring teams' tasks and deliverables in the project. Also tries to pitch-in as needed with other tasks and mainly to handle scrum meetings and feature design/architect roles in the team. He also has great deal of experience in developing through agile methodologies and infrastructure knowledge which can be leveraged for backend development and project deployment based on his previous experience as Unix Application developer.

Vidhi Bhatt
Vidhi will be working to help meet front-end and back-end requirements, make sure of, and take proper documentation such that project requirements and design concerns are being met. She also plans to assist in risk management and a little bit of backend mitigation. She has experience in front-end, little back-end, and little testing experience. She is available to take on testing roles in case testing leaders ask her. She is open to taking on more roles as needed as the project progresses.

Yasmeen Haleem
Yasmeen will be working with Gantt charts, documentation, and presentation. and plan to aid in usability testing and risk management and any services needed with integration with cloud. Along with these will assist in web app development and chatbot and other tasks assigned by different leads. She will be the documentation lead and will contribute to system testing and usability testing and will assist in risk monitoring, risk reevaluation and risk analysis and contingency plans to mitigate risks and will work with team for front end and back-end development.

Bhavani Rachakatla
Bhavani will help with the backend development and will also serve as the Testing Lead. She previously worked as an ERP application developer. She is also experienced in Machine Learning and will investigate the ML aspects of our project along with Yasmeen. She will also help with the documentation and presentation going further.

Demir Altay
Demir will help with the back-end requirements and design lead. Demir has previous experience in backend development with Python and Go as with GCP, our group felt he would be well suited to develop the backend.

Manushree Buyya
Manushree will help with the back-end development web-frameworks, assist in documentation, configuration according to requirements. She has experience in python and basic experience in front-end, integration, server and hosting so can help with this too according to the progress of the project. She has knowledge of data warehousing and business intelligence from previous internship work experience.

Pravallika Bollavaram
Pravallika will assist with back-end development and work on documentation, presentations, and website wireframes. Pravallika has experience in both front-end and back-end development. Pravallika is also open to assisting with testing, as she has previous experience as an ETL Tester.

Below table summarizes the roles but everyone would try to work on both development and other tasks as required during the complete duration of the project cycle.

Role(s)	**Team Member Name**
Project Manager	Kalyan Cheerla (Lead), Emmie Abels (Co-lead)
Requirements Lead	Vidhi Bhatt, Yasmeen Haleem
Design Lead	Kalyan Cheerla, Bhavani Rachakatla
Implementation Lead (frontend)	Emmie Abels (Lead), Vidhi Bhatt
Implementation Lead (backend)	Demir Altay (Lead), Pravallika Bollavaram
Configuration Management Lead	Pravallika Bollavaram
Testing Lead	Bhavani Rachakatla
Documentation Lead	Yasmeen Haleem
Demo and Presentation Lead	Manushree Buyya
System Administrator	Demir Altay
Database Administrator	Manushree Buyya

Member Contribution for Deliverable 1

Name	**Contribution**
Emmie Abels	Completed the Project Concept and Platform sections of the report and created the GitHub directory structure. Also assisted in leading group meetings, discussing project risks during group meetings and took meeting minutes.
Kalyan Cheerla	Still assisting the team as necessary with their tasks and heading the team to drive conversations during meetings. Created the High-level design diagram and probable front-end, back-end and database visualizations for our project. Assisted the squad with the technology stack and doubts regarding integrating ReactJS with Python Flask/Django. And suggested my thoughts and views on the probable risks and other.
Vidhi Bhatt	Assisted in discussing contingency plans, and project outline through meetings. Also helped

	document risks monitoring assessments, as well as, presented in the initial presentation, and developed a project description
Yasmeen Haleem	Assisted in making Gantt charts, and documented top three risks and contingency plans and in video presentation and took part in meetings and planning and discussions
Bhavani Rachakatla	I have outlined the tasks and subtasks essential for the project's success and have prepared a Gantt chart that encapsulates the entire project plan. Have participated in team meetings and discussions.
Demir Altay	I assisted in writing the languages and technologies section as with suggested some thoughts for potential risks.
Manushree Buyya	I assisted in presentation and documentation.
Pravallika Bollavaram	I helped prepare the presentation and made changes to the report.