

COMP SCI 520 THEORY AND PRACTISE OF SOFTWARE ENGINEERING

MID POINT PROJECT REVIEW

SUBMITTED TO:

PROF HEATHER CONBOY

SUBMITTED BY:

NITI MANGWANI

PRAJWAL SONTI HARI GANESH

SHREYA BIRTHARE

SIMRAN MALIK

Video URL: https://drive.google.com/file/d/1eST0Efxk8RqQ1J3Uk8bKixfAaRSPHvPD/view?usp=drive_link

GitHub URL: https://github.com/Nitss10/Patient_tracker_system

1. Requirements

1.1 Overview:

The goal of the project is to create a Patient Tracker System, a web based portal designed to streamline the management of patient information and medical records. It serves as a centralized platform for doctors, healthcare providers and patients to access, update, and maintain accurate patient records efficiently and securely.

In today's world data is an important resource. With an abundance of data, it is important to store and manage the data efficiently. Digitization is the need of today's times. Digitization in healthcare is no exception. Having a digital identity for a patient helps organize a patient's medical history, enabling more personalized and effective treatments. The project aims to eliminate the inefficiencies and errors associated with manual paperwork in the health industry. By digitization, the application seeks to streamline the process of managing patient information, thereby reducing administrative burdens and improving overall workflow efficiency. The goal is to provide real-time updates in patient records to ensure that healthcare workers have access to the most recent and accurate information. This feature is crucial for making informed decisions and providing timely and effective medical care.

1.2 Features:

By incorporating the below features, the Patient Tracker System aims to ensure data security, accessibility, and usability for both patients and doctors.

- User Authentication and Security:
 - Secure user authentication and the option for password reset.
 - Implementation of security measures such as a strong password policy, session management and temporary lockout after repeated failed login attempts.
- Patient Portal Features:
 - Personal Details Management: Allow patients to update and manage personal details, including address, password, email, phone number, and profile photo.
 - Comprehensive Medical History: Enable patients to maintain a detailed medical history, including the ability to upload medical records with optional remarks.
 - Appointment Management: Provide patients with the ability to schedule appointments, including specifying the purpose of the visit and preferred doctor for consultation.
- Doctor Portal Features:
 - Advanced Appointment Management: Allow doctors to view and manage appointments, apply filters, and search for specific patients based on various criteria and keywords.
 - Patient Record Updates: Enable doctors to add and update patient data.
 - Secure Data Viewing: Ensure doctors can view the complete patient data history, including medical reports, while preventing the editing of previous entries to maintain data integrity.
- Data Integrity:
 - Implement a robust system to track all changes made to patient records, ensuring data integrity and accountability.
 - Provide the ability to generate comprehensive reports on user activity and data modifications for administrative purposes.

- Usability and Accessibility:
 - Ensure a user-friendly interface for both patients and doctors, with intuitive navigation and easy access to all essential features and information.
 - Enable seamless experience for both patients and doctors, allowing them to access the system conveniently whenever necessary.

1.3 Functional Requirements:

- **For Patients:**
 - As a patient, I want to update my personal details such as my address and phone number so that the healthcare providers can reach me easily when needed.
 - As a patient, I want to upload my medical records such as CT scans and blood test reports so that my doctor can have access to my complete medical history during consultations.
 - As a patient, I want to schedule an appointment with a specific doctor for a consultation on a particular date and time so that I can receive timely medical attention for my health concerns.
 - As a patient, I want to view my upcoming appointments and previous appointment history to keep track of my healthcare schedule and medical history.
 - As a patient, I want to be able to log out securely from the system to ensure the confidentiality of my medical information and prevent unauthorized access.
- **For Doctors:**
 - As a doctor, I want to view the list of patients scheduled for the day along with their appointment details so that I can prepare for the consultations and organize my schedule efficiently.
 - As a doctor, I want to filter patients based on specific keywords in their medical records to prioritize urgent cases and access relevant patient information quickly.
 - As a doctor, I want to view the complete medical history of a selected patient, including their previous consultations and medical reports, to make informed decisions during the current consultation.
 - As a doctor, I want to add new patient data, such as symptoms, diagnosis, and prescribed medication, to the patient's record to maintain an accurate and up-to-date medical history.
 - As a doctor, I want to view detailed patient records securely without the ability to edit previous entries to ensure the integrity of patient data and prevent any accidental or intentional modifications.
- **Shared User Stories:**
 - As both a doctor and a patient, I want the system to maintain data integrity and ensure the security of all patient information to comply with privacy regulations and maintain confidentiality.
 - As both a doctor and a patient, I want the system to provide a user-friendly interface with intuitive navigation and easy access to essential features for seamless interaction with the platform.

1.4 Non Functional Requirements:

- Data Security and Privacy:
The model can support the implementation of secure user authentication and data encryption to ensure the confidentiality and privacy of patient data in the Patient Tracker System.
- Reliability and Availability:

By providing prompt and accurate responses to user queries, the model contributes to the reliability and availability of the Patient Tracker System, enhancing the overall user experience and ensuring continuous access to critical information.

- **Scalability and Performance:**

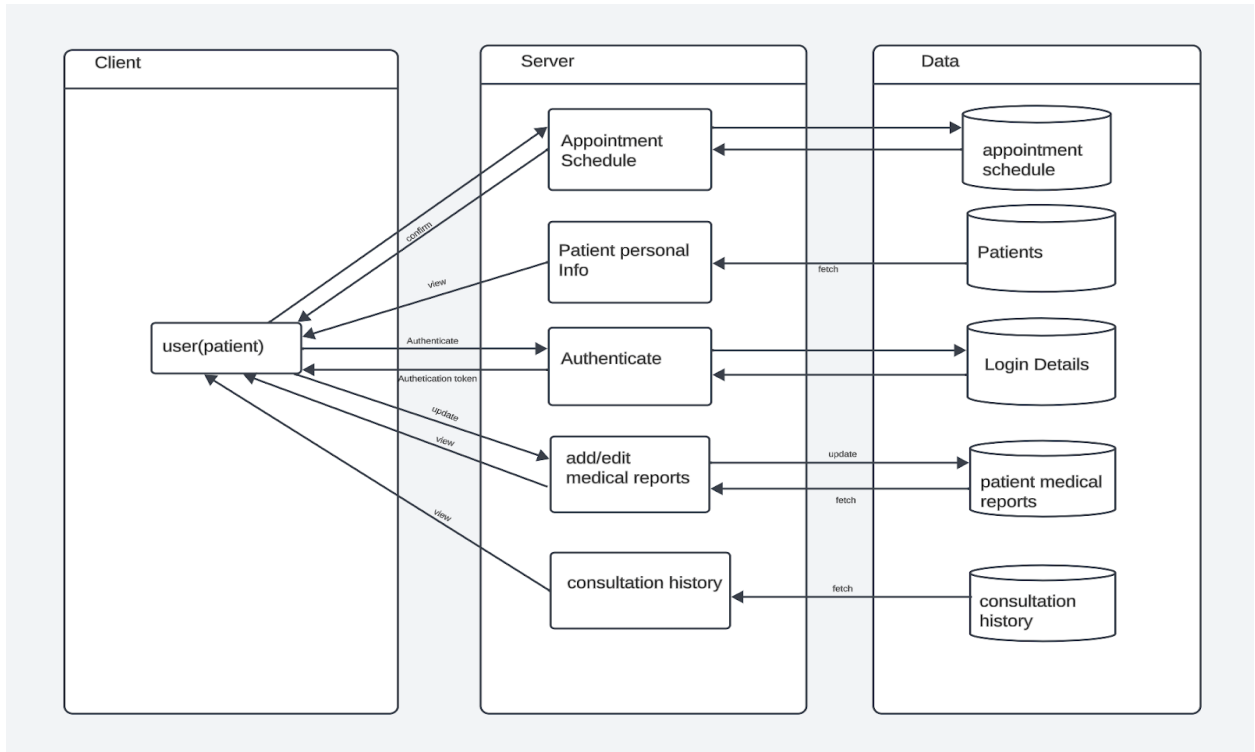
The model's ability to handle a variety of patient queries and provide real-time assistance contributes to the scalability and performance of the Patient Tracker System, allowing for efficient data processing and a seamless user experience even during peak usage times.

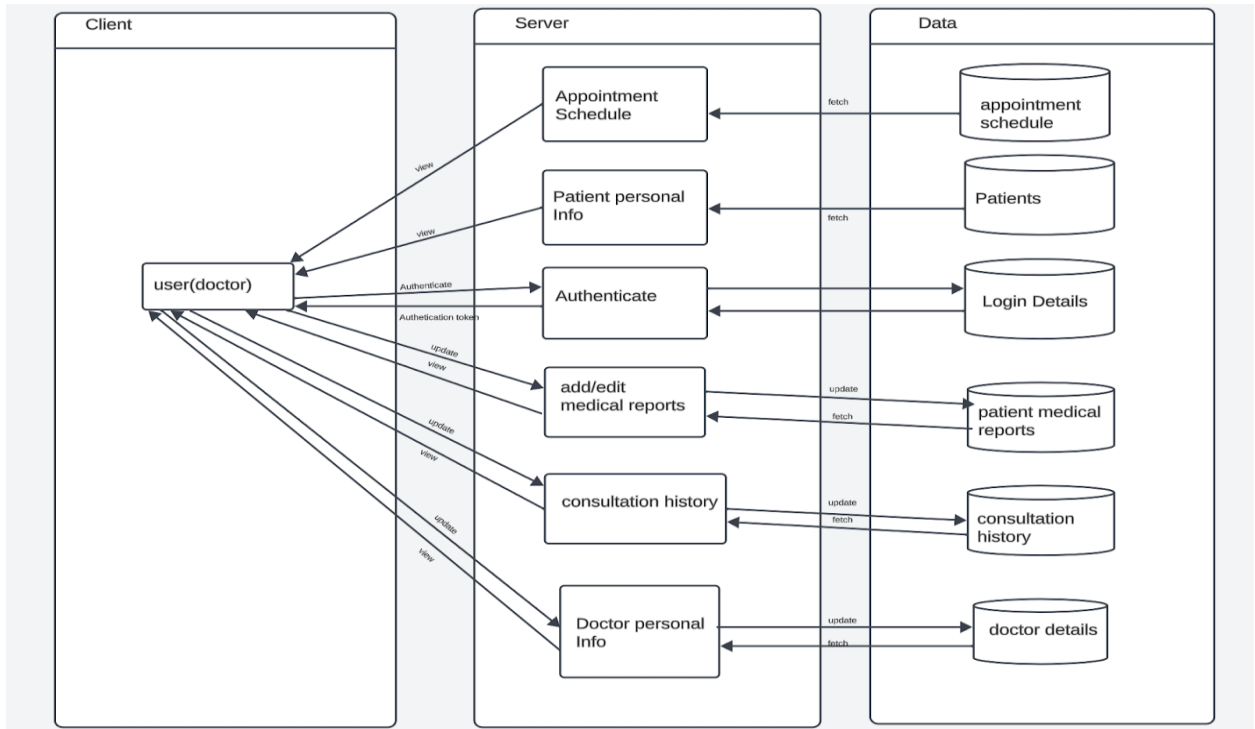
- **User Interface and User Experience:**

Integrating the model can enhance the user interface of the Patient Tracker System by providing intuitive and natural language-based interactions, improving the overall user experience for both doctors and patients.

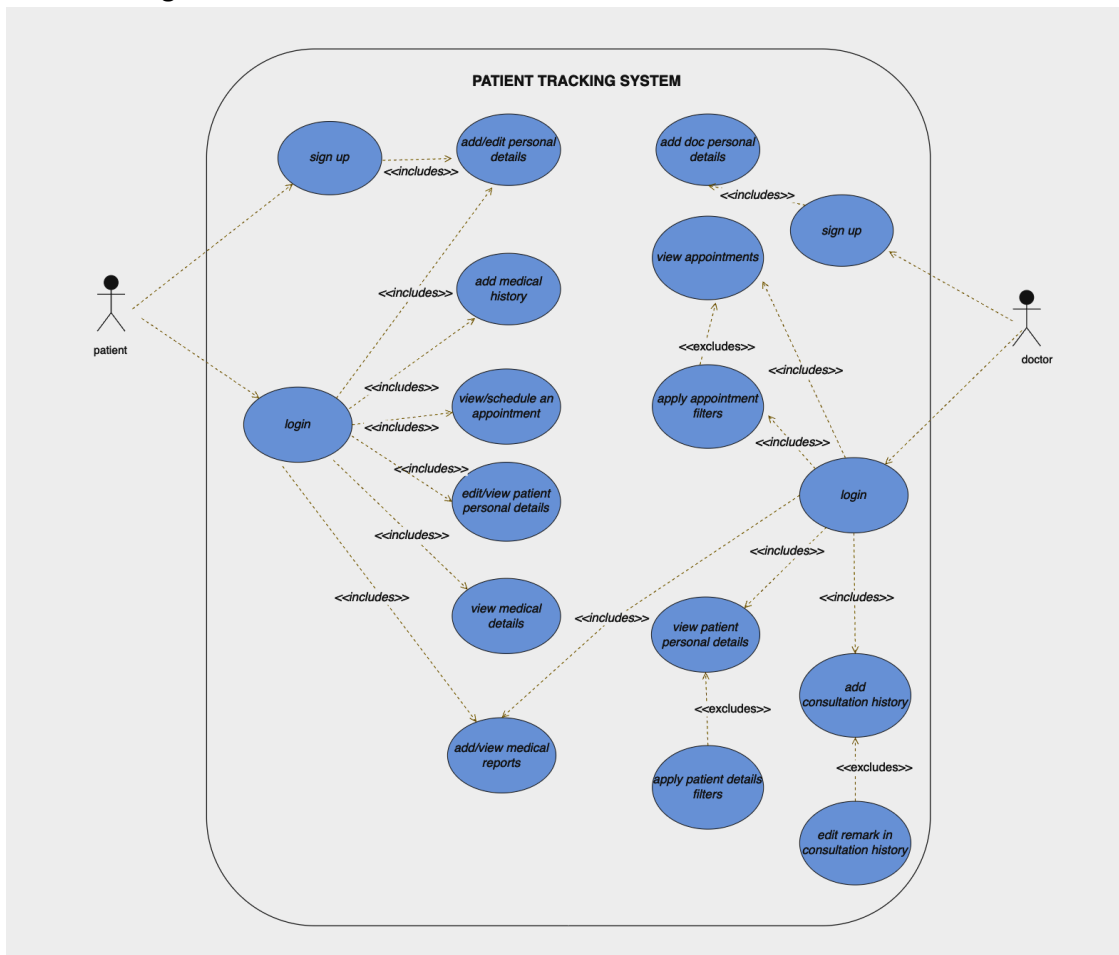
2. Design

2.1 Architecture Diagram:





2.2 Use Case Diagram:



2.3 Technology Stack with Justification:

The tech stack that we plan to use for this project is Django (Object Oriented python based framework) along with MYSQL for the database.

Django:

We are using Django because:

- The web framework Django offers capabilities and tools for both frontend and backend development, or full-stack development. As a result, integrating several technologies or frameworks is less necessary, which makes the development process more streamlined and effective.
- Django Web framework includes many built-in features for frontend development. It comes with its own template engine, Django Template Language (DTL), which makes it easy to generate HTML pages.
- Using Django for both backend and frontend development promotes consistency in your codebase.
- Django also includes an Object-Relational Mapping (ORM) system for database operations, built-in authentication, and a robust admin interface. These features can help streamline backend development and data management, which can be beneficial for the frontend.

MySQL:

We are using MYSQL because:

- Our patient tracker system requires the management of structured data, including patient details, medical history, appointments, and other structured information. MySQL, as a relational database, excels in handling structured data with predefined schemas, making it a good fit for applications like this.
- We desire data integrity and accountability, as well as the need to track changes made to patient records. MySQL is known for its strong support for ACID transactions, which ensures data consistency and integrity. It also allows the implementation of a robust system for tracking changes and maintaining data accuracy.
- Also MySQL's capabilities are well-suited for handling complex queries, which is important for generating comprehensive reports on user activity and data modifications. This is valuable for administrative purposes and data analysis.
- We would need security measures, such as secure user authentication and session management. MySQL provides robust security features and has a history of being used in applications that require strict security measures. You can implement a strong password policy and secure authentication mechanisms effectively.
- MySQL strongly adheres to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and consistency.
- MYSQL is highly compatible with Django. Django provides an Object-Relational Mapping (ORM) system that allows developers to work with databases using Python classes and objects, rather than writing raw SQL queries. This ORM abstracts the underlying database and provides a high-level API for database operations.

2.4 UI Mockups:

Landing Page:

PATIENT TRACKER

Doctors

Patients

Patient Sign in/Sing up Page:

USERNAME

PASSWORD

Sign In

Don't have an
account?

Register Now!

Account Creation:

Enter your Personal Information

First Name

Last Name

Email ID

Password

Confirm Password

New Patient Details:

Please enter your details below

Name

Address



Mobile Number

Age

Gender

Blood Group

Height

Weight

Medical Details

Basic Symptoms

Previous Diagnosis

Consulting Department



Doctor Name

REMARKS

Upload Reports

Returning Patient View:

Doctor Visits

•

•

•

•

Upcoming Appointment

Prescribed Medications

Reports

•

•

•

•

• Load more...

SCHEDULE APPOINTMENT

Purpose of Visit


Consulting Department

Severity

Doctor Name

Appointment Time

Available Time Slots

A rectangular button with a red header "Available Time Slots" and a black and white checkerboard pattern below it.

Doctor Main View:

Upcoming
Appointments

Date Patient Name Severity



Patient Lookup

SEARCH



Patient Directory from Doctor View:

Patient Directory

Name

Age: Gender:

Last Visit:

Latest Report:

Remarks

Previous Visits

•
•
•
•

• Load more...

Medications Prescribed

SEARCH

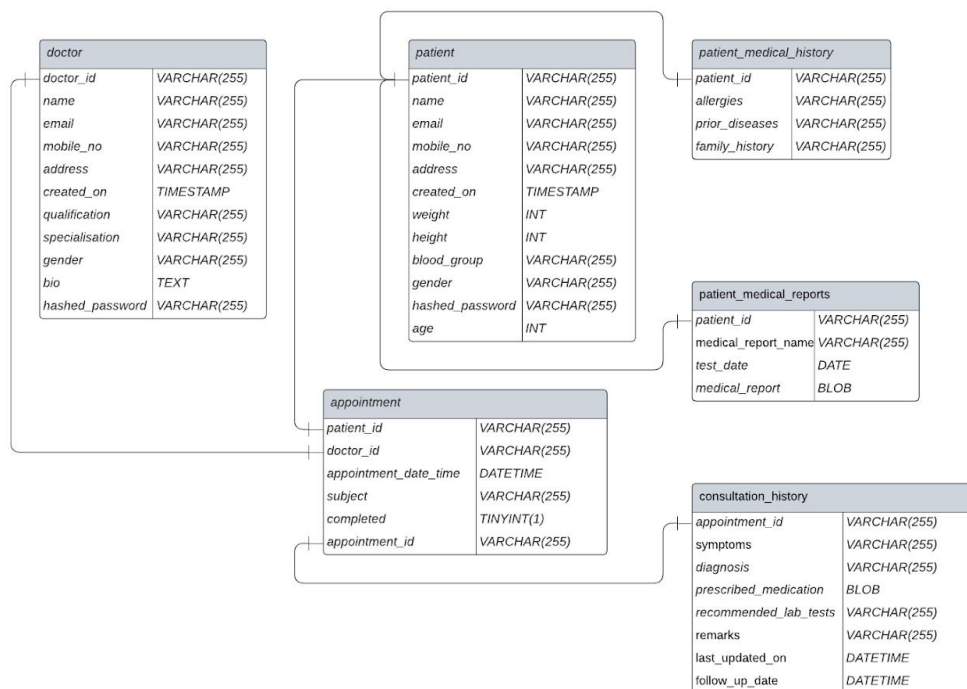
Enter your Personal Information

First Name	Last Name
Email ID	
Password	
Confirm Password	
Address	Age
	Gender
Mobile No.	Bio
Specialisation	Qualification

2.5 Data Model:

We will be using the MYSQL database for this Patient Tracker System. Below is the database diagram for the same system:

Database Diagram for Patient Tracker System



3. Implementation

Coding Process and Methodologies:

1. Requirements Gathering:

We started by thoroughly understanding the project's requirements, including user stories and non-functional requirements. This involved exploring different healthcare systems to ensure that our system meets its needs.

2. Architecture Design:

Next, we prepared a system architecture, specifying user authentication, data storage, and user interfaces. Used a layered architecture that separates concerns, ensuring scalability and maintainability. Also the architecture provides good access control functionality.

3. Database Design:

Brainstormed and created a robust database schema for storing patient information, appointments, and patient information including medical records. Sensitive fields implement necessary security measures like encryption and decryption.

4. Implementation:

We plan on developing the system using an object-oriented programming language like Python and will be using the framework Django. We will ensure that user authentication follows best practices, and data is securely stored and transmitted.

5. User Interface Design:

We will design a user-friendly interface for patients and doctors, focusing on usability and accessibility. Currently we have designed mock ups for the design that we plan on achieving.

6. Testing:

Once development is done, we will perform comprehensive functional testing and user acceptance testing to ensure the system functions correctly.

Potential Challenges:

1. Security: Ensuring data security is paramount. Protect the system against data breaches and unauthorized access. This requires robust encryption, secure authentication, and access control mechanisms.

2. HIPAA Compliance: Healthcare systems must adhere to HIPAA regulations to protect patient information. This involves strict data handling and privacy measures, as well as regular audits and compliance checks.

3. Scalability: The system should be able to handle a growing number of patients and doctors without performance degradation. Scalability challenges may arise as the user base increases.

4. User Adoption: Healthcare professionals and patients may be resistant to change. Ensuring a user-friendly interface and providing adequate training and support can mitigate this challenge.

5. Data Integrity: Preventing accidental or intentional data corruption is crucial. Implementing an audit trail and ensuring that previous entries cannot be edited is essential for data integrity.

6. Legal and Ethical Considerations: Complying with legal and ethical standards, beyond HIPAA, may present challenges. For example, dealing with data ownership, consent, and ethical data use.

Security and Risks:

1. Data Breach:

- Potential Threats and Vulnerabilities: Threats include unauthorized access, hacking, and insider threats. Vulnerabilities may be weak authentication, unencrypted data, or improper access control.
- Measures to Prevent Unauthorized Data Access: Implement strong user authentication, data encryption, access control, and regularly update security protocols.

2. HIPAA Compliance (in case of Patient Tracker):

- Explanation of HIPAA: HIPAA is a U.S. law that sets standards for protecting sensitive patient data, including electronic health records (EHRs). It mandates strict security and privacy measures for healthcare systems.
- Relevance to the Project:
HIPAA is highly relevant to a Patient Tracker System because it involves the management of sensitive patient health information. Failing to comply with HIPAA can lead to severe legal and financial consequences, including fines and penalties. It's crucial for the project to protect patient data, maintain its integrity, and ensure that only authorized personnel can access it.
- Steps and Protocols to Ensure Compliance:
 - We will ensure that all patient data, both at rest and in transit, is encrypted. This includes encrypting data while storing in databases and decrypting it when needed for use.
 - We will use strong encryption protocols, such as TLS for data in transit and encryption at the database level for data at rest.
 - We will implement user authentication and access controls to limit data access to authorized personnel.
 - We will collect and retain only the minimum amount of patient data necessary for the system's purpose. Avoid collecting extraneous data that is not needed.
 - We will implement strong user authentication, including unique usernames and strong passwords. We might consider adding Multi-Factor Authentication (MFA) for added security.
 - We will ensure that all personnel handling patient data are trained on HIPAA requirements and are aware of their responsibilities in maintaining data security and privacy.
 - Ensuring data security and HIPAA compliance requires a combination of technical measures, user training, and legal awareness to protect patient information and maintain the integrity of the Patient Tracker System.

Work Plan

High-level Timeline:

We are following the sprint based agile development model. Each of our Sprint is approximately 15 days. In Sprint 1, we identified the functional and non-functional requirements of our project, created an architecture diagram and decided on what tech stack to use. In the second sprint, we dived deeper into the technicalities of the project and decided the tech stack and the low level design and workflow of the project. Furthermore, we also created the midpoint review document and recorded our video and present the project to our peers. In the next sprint, we are going to start developing the individual components. This should be completed by the second week of November. In the final Sprint, we are going to perform rigorous testing of our application and identify any bugs or loopholes in the application from a working and a security standpoint. We plan on completing the project by the first week of December.

The project is going to be split among the team members with each team member working on a major chunk as follows: Shreya and Simran are going to start with the backend development, Niti is going to start with the database design and modelling and Prajwal is planning to work on the front end of the project. Depending on further bandwidth the work will be distributed equally among team member. We will be use version controller tools like git for pushing and tracking the changes in the project and will be performing some level of testing to ensure the project is working as expected.