

**Envo Club**

**Founder**: Saransh Hasija

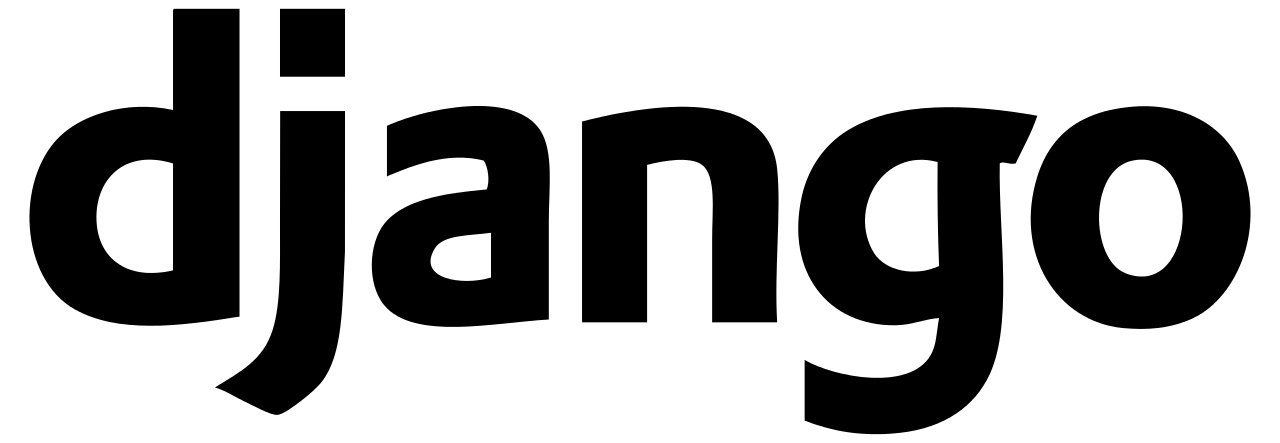
**Project Manager**: Sourijit Das

**Full Stack Developer**: Sarthak Punjabi

**Sustainability Lead**: Raksha Raina

**Brand Partnerships**: Dhruv Kabra

**Social Media Lead**: Lydia Collins



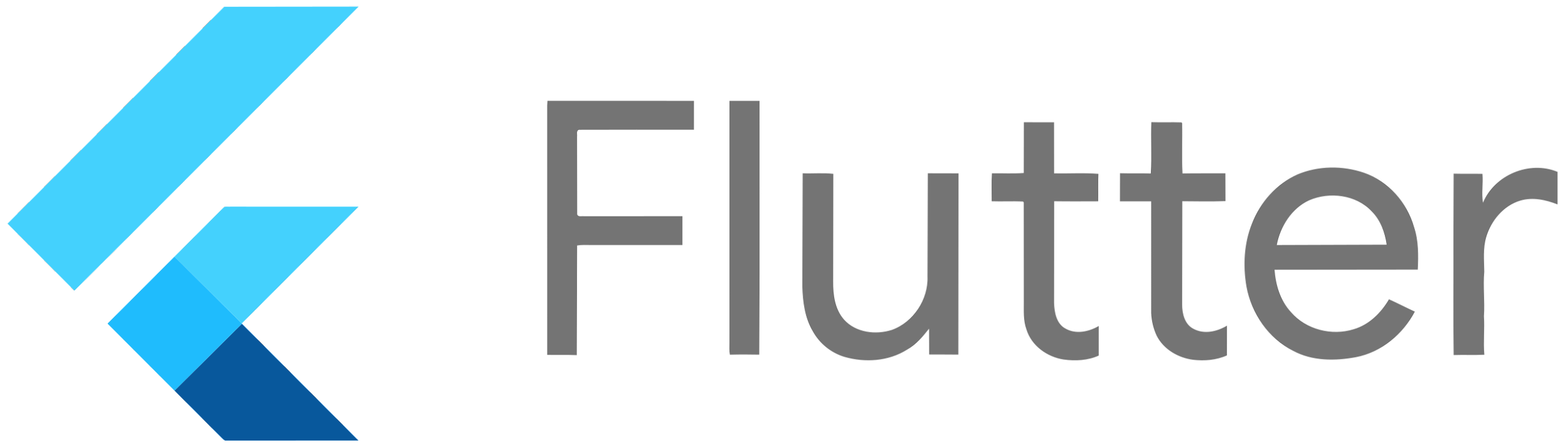


Table of Contents

1. Project Description

1.1 Overview

1.2 Business Scenario

1.3 Case tools

1. Software Lifestyles and Methodologies

2.1 Agile-Scrum

2.2 Click Up

2.3 Github

2.4 AWS

1. Project Plan

3.1 Team Role Declarations

3.2 Work Plan

1. Requirements

4.1 Functional Requirements

4.1.1 Use Case Diagram

4.1.2. Use case description

4.2. Non Functional Requirements

4.2.1. Security

4.2.2. Extensibility

4.2.3. Portability

4.2.4. Performance

4.3 GUI Prototypes

4.3.1. Home Screen

4.3.2. Social Media Screen

1. Architectural styles and patterns

5.1. Model View Template

5.2. Layer Architecture

5.3. Bloc Architecture

1. System Architecture

6.1. Package diagram

6.2. Technologies in use

6.3. Deployment details

6.3.1. Application Running on AWS - EC2 instance.

6.3.2. Apache Web Server Configuration

6.3.3. Jenkins Pipeline

6.3.4 Requirements.txt

1. Sketches of Analysis artifacts

7.1. Class diagram

7.2. Communication diagram

7.3. Sequence diagram

7.4. Entity Relationship diagram

1. Lines of code
2. Code - Design Patterns

9.1. Factory

9.2 Decorator

9.3 Singleton

9.4 State Pattern

9.5 Observer Pattern

9.6 Pytest - Unit Test Cases.

9.6.1 Unit Test for Models

9.6.2 Unit Test for Views

9.6.3 Unit Test Cases for Urls

9.6.4 Unit Test Case Result

9.7 Flutter Test - Frontend Test Cases

1. Code – Added Values

10.1. DevOps - AWS and Jenkins

10.2 Django Rest Framework

10.3 Flutter

10.4 JWT Tokenization

10.5 SonarQube(pylint)

1. Recovered architecture and design blueprints

11.1 Architecture Diagram

11.2 Design-time class diagram

11.2.1 Python Django Design-Time Class diagrams

11.2.2 Python Django Design-Time Class diagrams

11.3 State Chart diagram based on sequence diagram

1. Component & Deployment Diagrams UML
2. Critiques
3. References Cs
4. Appendix : Weekly diaries

1. Project Description

1.1. Overview

Rentezzy allows a customer to search and rent property in Ireland.​ This platform has 3 types of users - customer, owner and agent. Owner and agent can add property to be rented. Agent has to have a subscription for posting properties and in turn can earn commission out of it, whereas the owner can sign up and add property available for rent. Customers will be able to book and also cancel the booking after a certain amount of time. Rentezzy also allows users to get details of the property location from important stores, transportation, etc.

1.2. Business scenario

We are required to deliver a system that allows how a customer may search for a property and book property for rent. The system checks the desired properties from the database based on counties and provides results to the customer. Also we have agents as another user of this system where they can add property that is available for rent. Agents can get commission based on properties that are booked amongst the listed ones. Agents can be promoted to gold and platinum based on the properties rented that are listed.

1.3. Case tools

**Intellij PyCharm:** Intelij PyCharm is the IDE (integrated development environment) used for code development. Intellij PyCharm (Community edition) is a free IDE available from <https://www.jetbrains.com/pycharm>.

**InVision:** Invision is the tool used for designing UI-UX Prototypes. It allows users to create interaction among the elements and different pages.

**Visual Studio Code :**It is an IDE used for code arrangements and development in multipurpose language python and Component based Flutter for UI

**Lucid Charts:** we have used lucid charts for making the UML Diagram

2. Software lifecycle and Methodologies

2.1. Agile-Scrum

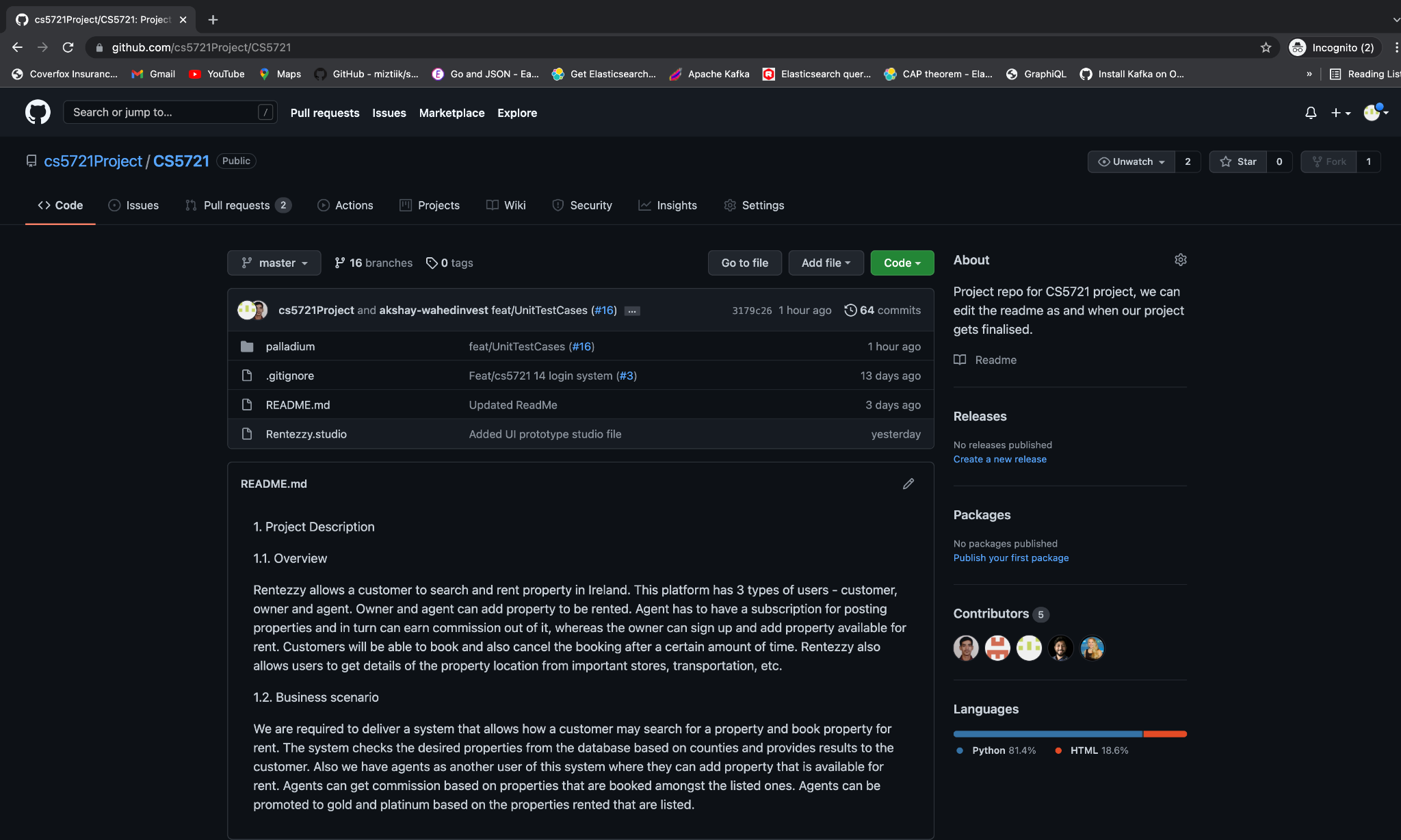
We have used Agile methodology to develop this project. We found agile to be efficient as it focuses on developing features. i.e. use cases we have planned so far and also it is an iterative process so we have divided our work in sprints. We have used Scrum as our agile framework. The Sprint was time bounded by 14 days and we had sprint planning and retrospective meetings on the first and last day of the sprint respectively. Quarter weekly stand-up meetings were scheduled in the team to discuss the progress – What is done, what is the next plan and impediments if any. If any of the team members were blocked by any issues, meetings were arranged to resolve the issue.

2.2. ClickUp

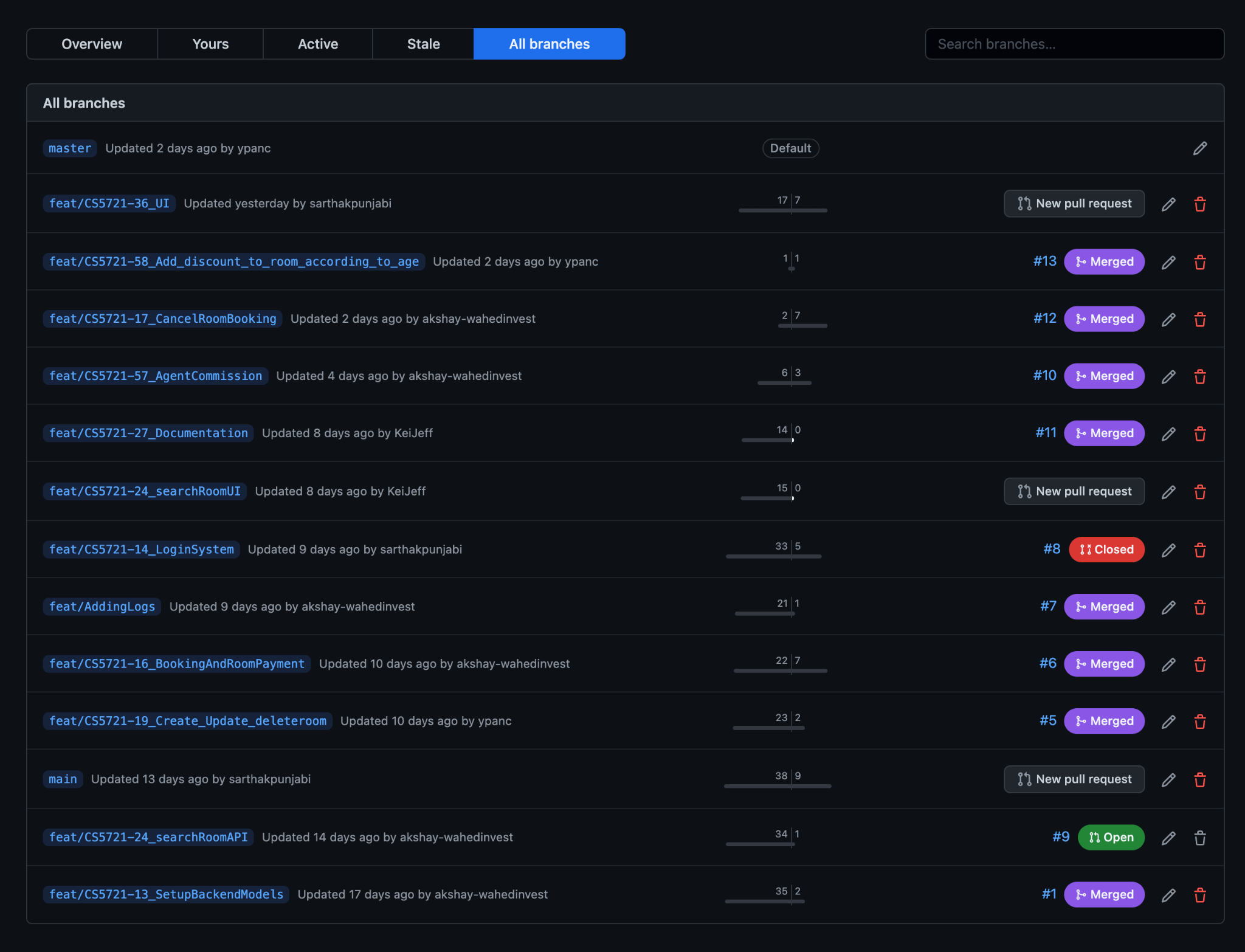
We have used “Click Up” Software to maintain the status of features, tasks that are planned in the sprint. We have 3 stages to make the task to the final stage that are – To do, In progress and done. Communication was done through ClickUp and MS Teams.

2.3. GitHub

Git was used to manage the code version. Team members created separate branches based on features and pushed the code in the same branch. Pull requests were made by team members in order to get it reviewed by another team member before merging the code into master to enable Continuous Integration and Continuous Deployment.

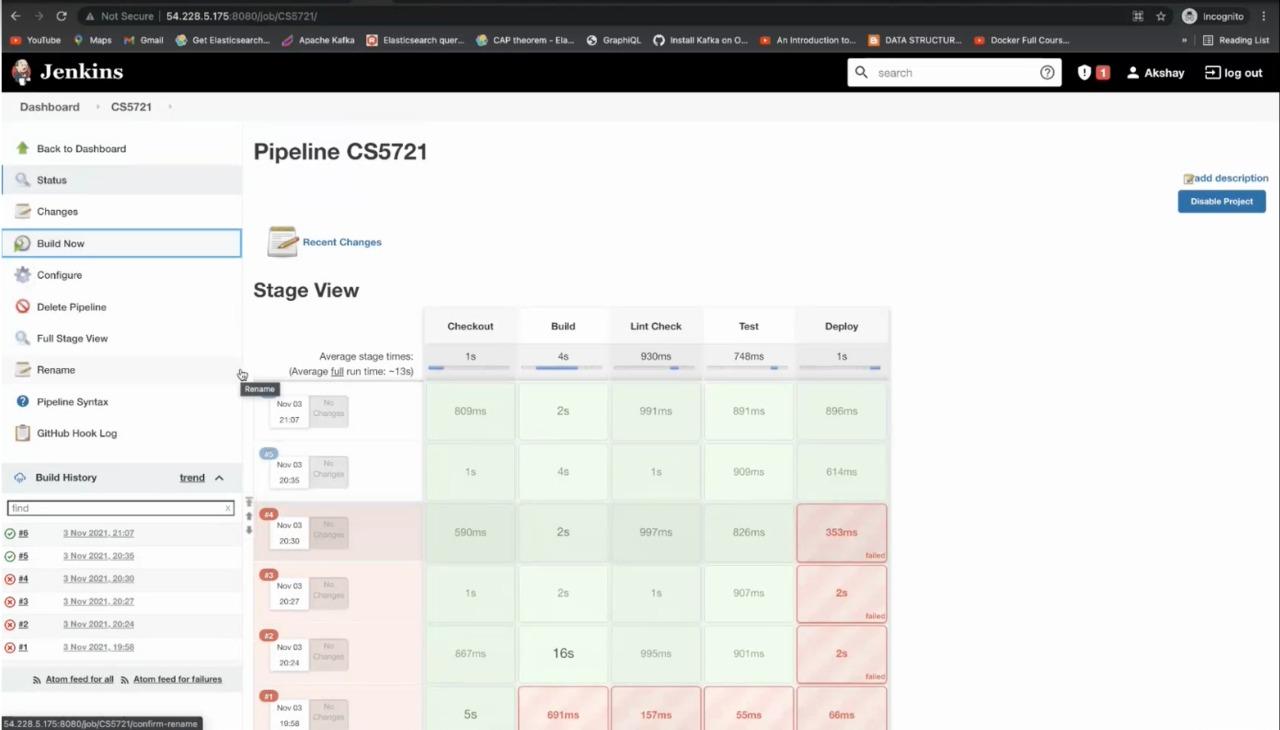


Below is the snapshot of our branch created for feature development, we have followed best practice for naming the branch i.e. feat/<Clickupno>\_ <StoryName>



2.4. Jenkins

We have used **Jenkins** for automating the build, test and deploy of the code. Jenkins is an open source automation server which helps in CI/CD. We have integrated Jenkins with AWS for CI/CD.



3. Project Plan

3.1 Team Role Declarations

| **Sr No.** | **Role** | **Description** | **Designated Team Member** |
| --- | --- | --- | --- |
| 1 | Project Manager | Sets up group meetings, gets arrangements on the project plan and tracks progress. | Sourojit Das |
| 2 | Documentation Manager | Responsible for sourcing relevant supported documentation from each team member and composing it in the report | Saransh Hasija |
| 3 | Business Analyst/ Requirement Engineer | Responsible for section 6 - Requirements | Dhruv Kabra  Sarthak Punjabi |
| 4 | Architect | Defines system architecture | Sourijit Das |
| 5 | Systems Analysts | Creates conceptual class model | Sarthak Punjabi |
| 6 | Designer | Responsible for recovering design time blueprints from implementation | Sarthak Punjabi |
| 7 | Technical Lead | Leads the implementation efforts | Sourijit Das |
| 8 | Programmers | Each team members to develop at least 1 package in the architecture | Sarthak Punjabi |
| 9 | Tester | Code for the automated Test cases | Sourijit Das  Saransh Hasija |
| 10 | Dev Ops | Must ensure that each team member is competent with development infrastructure. | Sarthak Punjabi |

3.2 Work Plan

| **Week** | **Workflow** | **Output** |
| --- | --- | --- |
| 3 | Setup team roles | Roles identified based on strengths of each member |
| 4 | Requirements | Requirement analysis performed and core idea of the project conceived. |
| 5 | Analysis | Analysis on the scope of the project and initial use case diagrams drawn. Identified the business logics |
| 6 | High level architecture | Team leader and Architect discussions on architecture, language and pattern of application decided |
| 7 - 12 | Coding literation | | First | Setup DevOps, Jenkins Pipeline with SonarQube (pylint), Clickup, GitHub repository and finalise UML diagrams. | | --- | --- | | Second |  | | Third |  | | Fourth |  | |
| 13 | Testing | Continuous testing was performed using pytest testing |
| 7 | Deployment | Continuous deployment was done using Clickup CI to integrate the working code. |