Table of Contents

[Table of Figure 3](#_Toc124538359)

[Introduction 4](#_Toc124538360)

[Tools and Technologies 5](#_Toc124538361)

[HTML 5](#_Toc124538362)

[CSS 5](#_Toc124538363)

[JavaScript 5](#_Toc124538364)

[Bootstrap 5](#_Toc124538365)

[Python 5](#_Toc124538366)

[Django Framework 5](#_Toc124538367)

[Django-Jazzmin 5](#_Toc124538368)

[Version Control 6](#_Toc124538369)

[Git Workflow 6](#_Toc124538370)

[Database: 9](#_Toc124538371)

[Automated testing 10](#_Toc124538372)

[Unit testing( TDD) 10](#_Toc124538373)

[Integration testing (BDD) 10](#_Toc124538374)

[Problem Faced 10](#_Toc124538375)

[Difficulty in setting up testing environment: 10](#_Toc124538376)

[Difficulty in testing views and templates: 11](#_Toc124538377)

[Difficulty in testing third-party packages and apps: 11](#_Toc124538378)

[Conclusion: 20](#_Toc124538379)

[References 21](#_Toc124538380)

# Table of Figure

[Figure 1: GitHub Organization as Sipalu 6](#_Toc124451685)

[Figure 2: Master branch as major branch with all merged branches 7](#_Toc124451686)

[Figure 3: All Branches 8](#_Toc124451687)

[Figure 4: Sipalu Database 9](#_Toc124451688)

[Figure 5: Ran Test 11](#_Toc124451689)

[Figure 6: Testcase for Login 12](#_Toc124451690)

[Figure 7: Testcase for Profile 13](#_Toc124451691)

[Figure 8: Testcase for Category and Services 14](#_Toc124451692)

[Figure 9: Testcase for Order 15](#_Toc124451693)

[Figure 10: Testcase for BlogComment 16](#_Toc124451694)

[Figure 11: Testcase for ShippingAddress 17](#_Toc124451695)

[Figure 12: Testcase for OrderItem 18](#_Toc124451696)

[Figure 13: Testcase for Checkout 19](#_Toc124451697)

# Introduction

As part of the final assignment of the Agile Development module, our team, The Bhabbyezz, developed a web application named Sipalu. Sipalu is a web platform for freelancers and skilled workforce of Nepal to sell their skills and services online to take part in the gig economy of Nepal. Gig economy can be understood as the work market characterized with the involvement of short-termed deals and contracts in contrary to traditional job concepts. The Sipalu application is a web-based application powered by Django backend framework. On the client-facing portion, various technologies such as HTML, CSS, JavaScript, and Bootstrap were implemented. Along with these backbone technologies, we utilized the Jazzmin admin panel provided by the Jazzmin library in Python to construct a reliable admin panel for the application. Similarly, we utilized the Twilio chat bot to build a communication channel between the client and the admins of the application. The front end of the Twilio chat bot is displayed in the site as a pop-up chat icon and when the client opens the chat bot, s/he can easily send a message to the admins of the application. The fundamental project's repository, or code base, was hosted on GitHub. All of the team members joined a separate organization we called Sipalu when we formed it.

# Tools and Technologies

In order to built this web application various technologies and packages were implemented. Some important technologies used in the development of the Sipalu application is explained below:

## HTML

HTML was utilized to create the foundation and structure of all elements on the website. The initial visual layout of the website was created using HTML.

## CSS

Cascading Style Sheets (CSS) was employed to apply visual styles to the pages of the site. This allowed for precise control over the appearance and design of each component on the front-end of the application.

## JavaScript

Some JavaScript programming was done in the web pages in order to make them dynamic. A static page in an application such as Sipalu would seem dull and inappropriate.

## Bootstrap

We even utilized the ready-made components provided by the bootstrap framework. Thinking that it is unnecessary to “Reinvent the Wheel”, we utilized many pre-designed components of the Bootstrap framework such as the Forms, navbars, footers, etc.

## Python

The python programming language was used to program the backend functionalities of the Sipalu application. The easy to learn curve of the python programming language enabled us to implement our algorithms into syntax without much difficulties.

## Django Framework

The Django web framework of python was used to develop the backend of the Sipalu application. The easy to understand and use MVC model of Django architecture was helpful throughout the development process.

## Django-Jazzmin

The Jazzmin is a drop-in library for our web applications which provides us with pre-built admin panel that just needs an integration with our application in order to work. The Jazzmin admin provides all the generic admin functionalities demanded by most of the applications. We utilized the many useful pre-built functionalities of Jazzmin to empower the admin panel of our own application.

# Version Control

Every system under development, be it a small side hustle project or a huge project aimed at market release, the development lifecycle demands the project to be scrumbled up and divided into multiple versions of the system. On our project too, we developed the Sipalu application by developing the project version wise with each progressive version of the system having an additional feature. Necessary branches were created during the development timeline to develop new features into the system and finally, when the new branch’s features were developed, tested and refactored, the branches were merged.

## Git Workflow

For version control, we used git workflow to manage and collaborate on the source code among the development team. Developers use git to create branches for new features or bug fixes, which allows them to make changes in parallel without affecting the main branch. Each developer committed their changes to their own branch and then pushed them to the remote repository. Pull requests are used to review and approve changes before they are merged into the default development branch. Once changes are merged, the main branch is deployed to a test environment for further testing and validation before being deployed to production. By using git and a specific workflow, the development team can easily track changes, collaborate on code, and manage versions. It also enables them to maintain a history of the codebase and rollback to previous versions if needed.

The following figure is the organization that we created for our web application called Sipalu where our five members were involved.

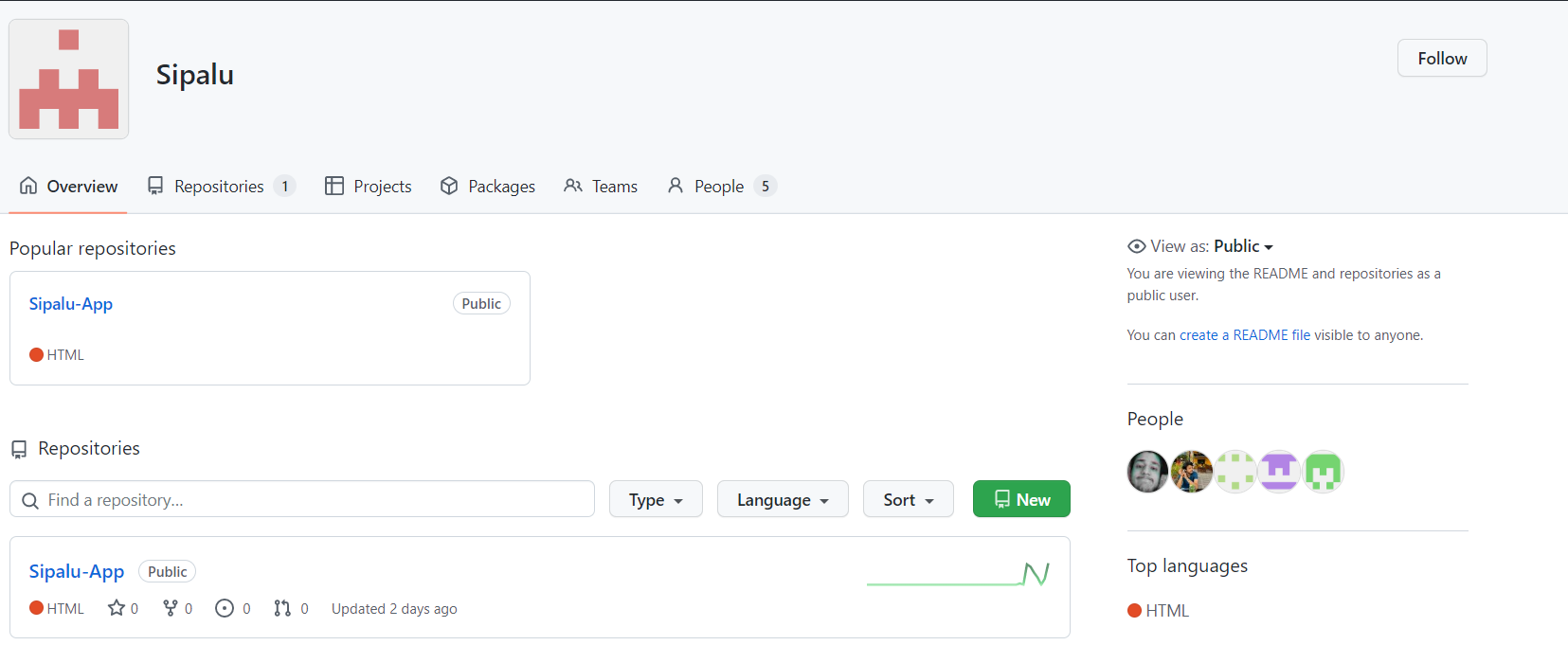


Figure 1: GitHub Organization as Sipalu

GitHub was used as the primary code base or the repository for the main project. We created a separate organization with the name of Sipalu and all the team members joined up into the organization. A repo was then initiated to act as the codebase for the application. All the team members pushed their work into the repository with each sprint.

The following figure shows the master branch which is merged within 10 branches with various features:

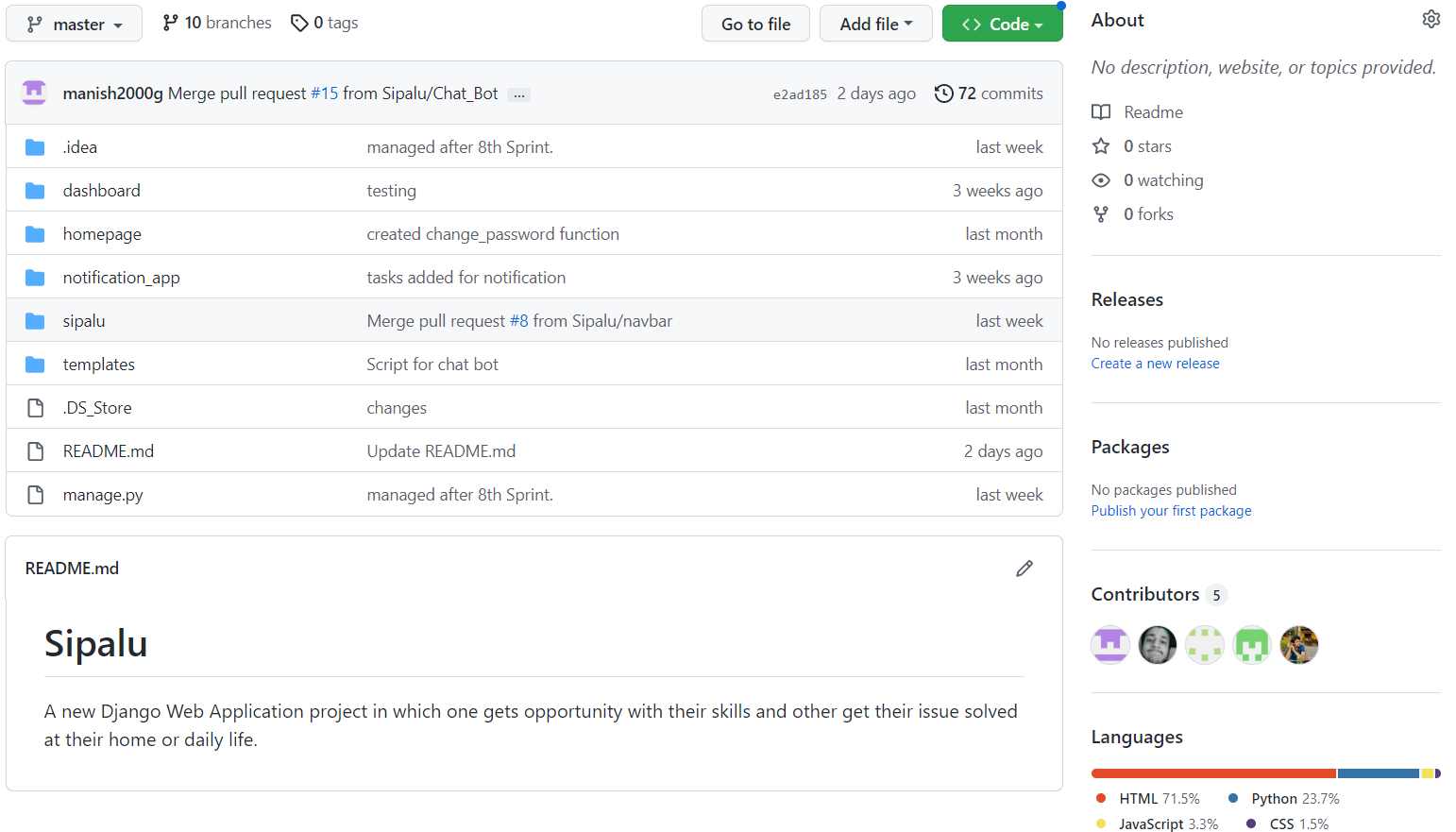


Figure 2: Master branch as major branch with all merged branches

Necessary branches were created as per the demand of the sprint like Navbar branch, Register branch, Category branch, Login branch, Notification branch and other four branches according to the features implemented along with default development branch Master.

All the branches are shown below in the figure:

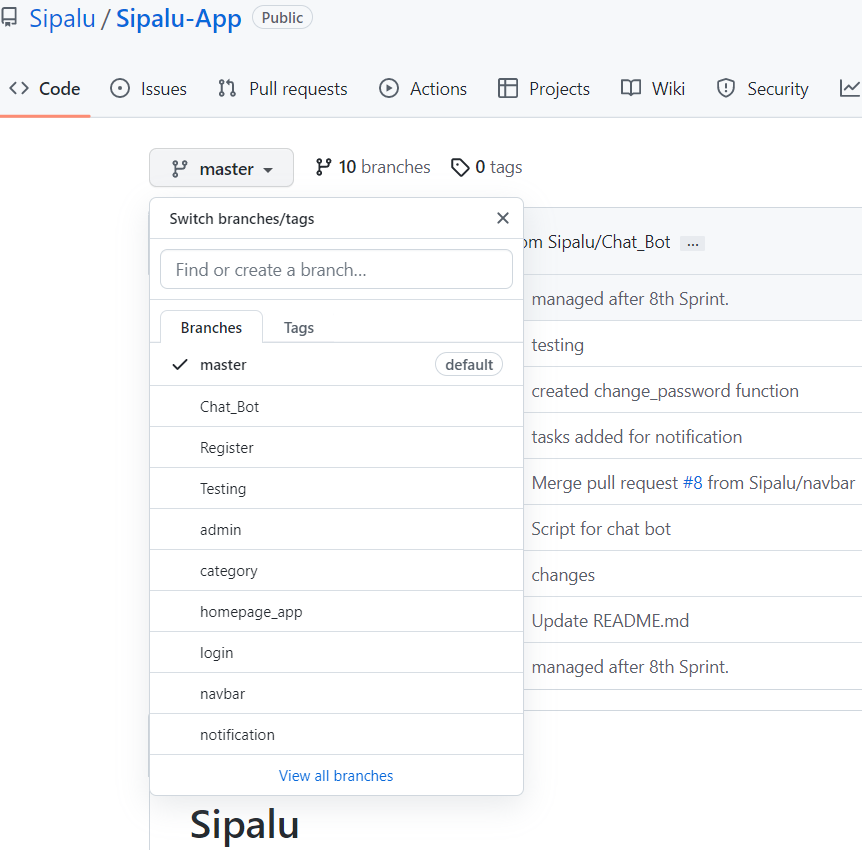


Figure 3: All Branches

Then, necessary update to the version of the application was updated and the code was pushed. Finally, the branches were merged to the main branch after verifying that the new version is bugless. All the team members contributed to the iterative development of the product with the help of Git version control system.

# Database:

Like all real-world applications, our application too would require a reliable database to store the client credentials as well as all the records of orders and transactions in the application. Many avenues to address the demand of a database were at hand. We, as a team, were confident with using both SQL as well as No SQL databases. We had already gained experience working with SQL, Mongo DB, Graph DB, etc. in the earlier projects too. However, we were in search for a more universal and easier to integrate with python kind of database for our application. This led to the selection of My SQL as the most reliable and user-friendly database for our application. Several tables were created to store the relevant data about the clients and transactions/orders. Similarly, separate categories of tables were required to store information related to the admin panel of the application. The Django framework as well as the python programming language is easily compatible with the My SQL connector. This helped us to flexibly play with our data and to store the data as per our necessities. The following figure shows the database table of Sipalu:

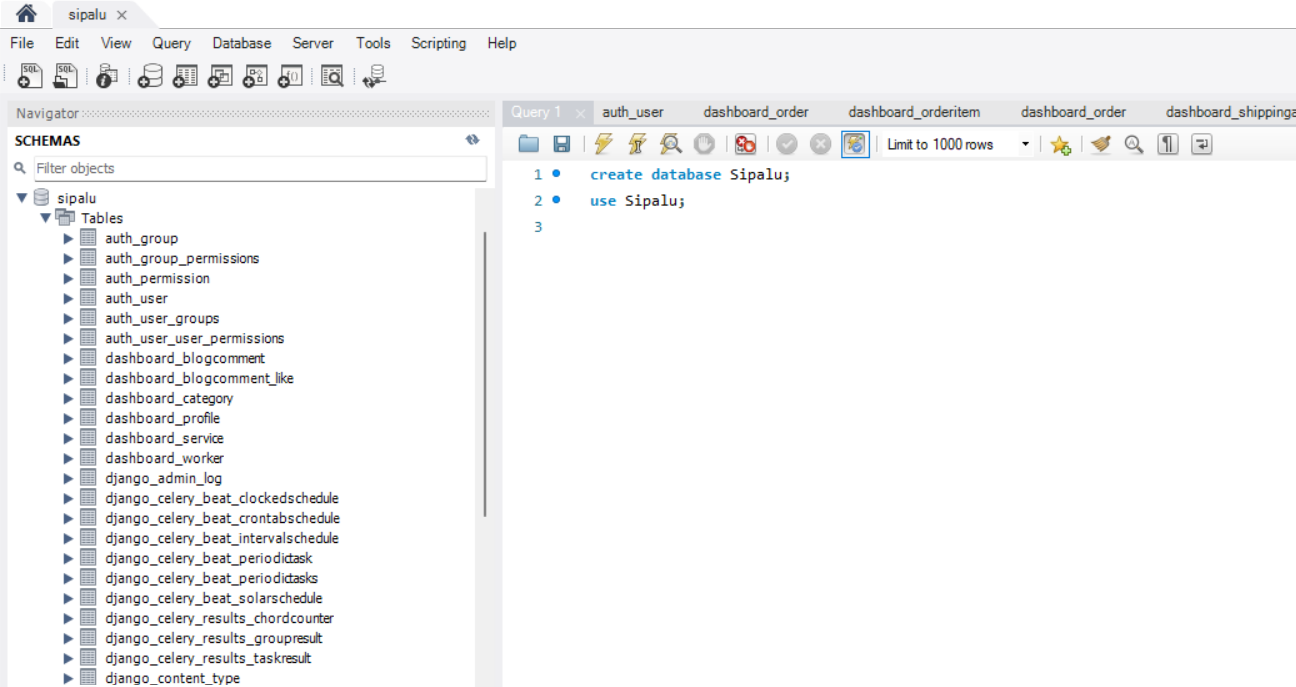


Figure 4: Sipalu Database

# Automated testing

Automated testing refers to a method of software testing where testing tools are used to verify that software is functioning correctly and meeting the specified requirements. It checks for errors, flaws, and any other problems that could occur throughout the creation of the product. It uses scripted process to test the software then the information or the result is reported. The results obtained can be compared with previous test runs.

Implementing automation technologies allows for more regular testing, which enhances functionality in general. Cycles of software development involve continuous testing, frequently the same test. It is made feasible via automation testing, freeing up team members for other tasks. Compared to manual testing alone, it can also produce results that are more precise and dependable. A further verification that the product is fit for the market or to go to the following development stage. This assurance encourages the team to keep improving.

Types of automation testing:

There are generally four types of automation testing. Each testing is done based on the circumstances and depending on the software that is being tested. Some of the important automation testing are given below:

## Unit testing( TDD)

Unit testing and Test-Driven Development (TDD) have a strong relationship. In unit testing, specific segments of code such as functions or methods are tested to guarantee they are functioning as expected. Conversely, Test-Driven Development (TDD) is a software development methodology in which unit tests are written before any corresponding code is written. The reasoning behind this is that by writing the tests first, developers will have a solid understanding of the desired outcome and can then write the production code to fulfill those test cases.

## Integration testing (BDD)

Both integration testing and BDD testing are concerned with how different parts of a system interact with each other, but integration testing focuses on testing the interactions between different components of the system, while BDD focuses on testing the interactions between the system and its users by creating scenarios that describe the desired behavior of the system. The big bang approach, bottom-up approach, top-down approach and the sandwich approach are some approaches to integration testing.

## Problem Faced

During the development of this project, several issues were encountered during the testing process. There are various challenges that can arise when using Test-Driven Development (TDD) and Behavior-Driven Development (BDD) testing methods. Some of the issues that were faced during its process are:

### Difficulty in setting up testing environment:

Since, we did our project using Django framework, we had trouble to set up test environment because the project had several dependencies.

### Difficulty in testing views and templates:

In Django framework, it’s views and templates rely on request and response object. This made us difficult to test them.

### Difficulty in testing third-party packages and apps:

We have used Twilio, a communication API for SMS, voice, video, etc. in our program so testing third party app was really a hefty task.

However, these issues were overcome by using right tools and coordinating with the team members. Django offers a built-in test framework as well as different third-party libraries which can be beneficial while testing views, templates and other components of Django project.

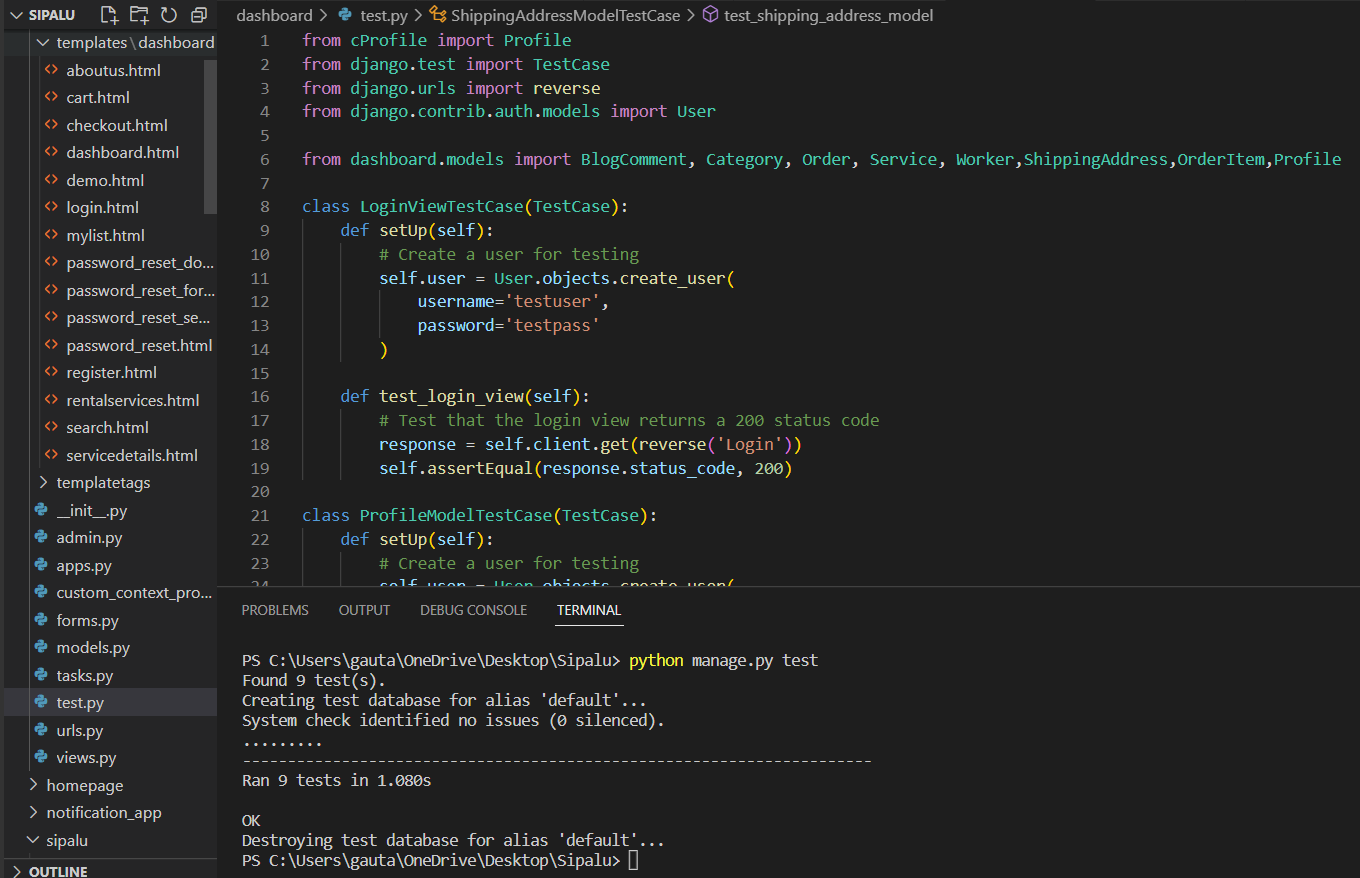


Figure 5: Ran Test

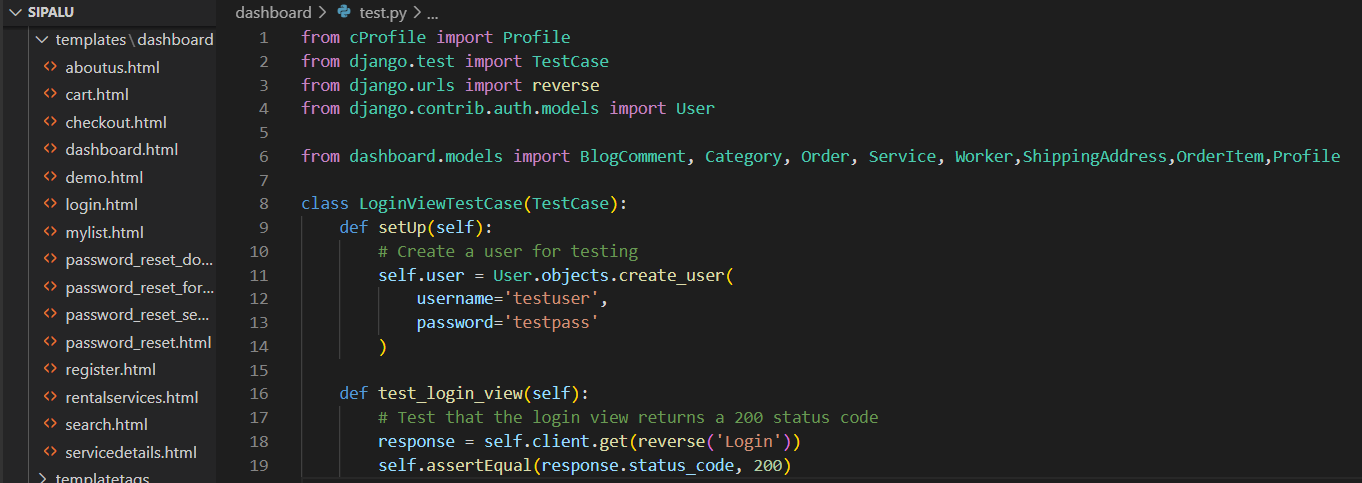


Figure 6: Testcase for Login



Figure 7: Testcase for Profile

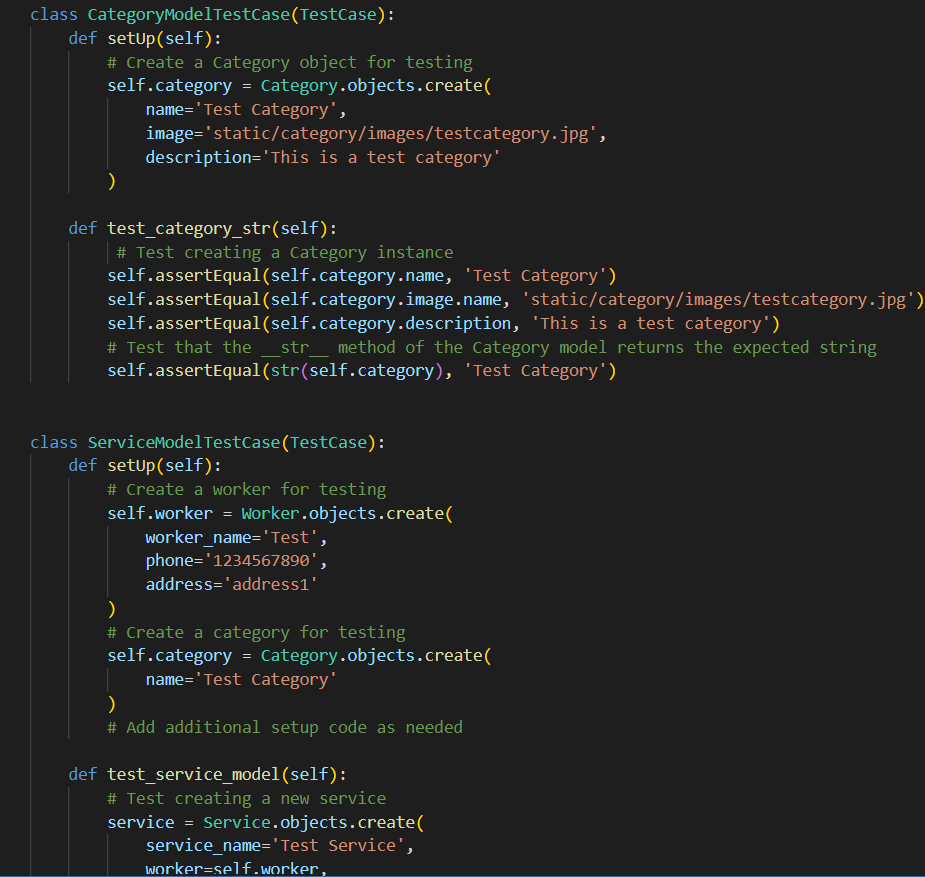


Figure 8: Testcase for Category and Services

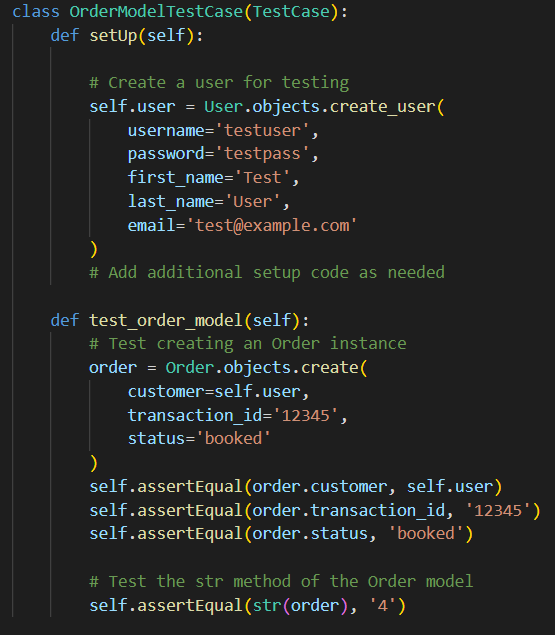


Figure 9: Testcase for Order

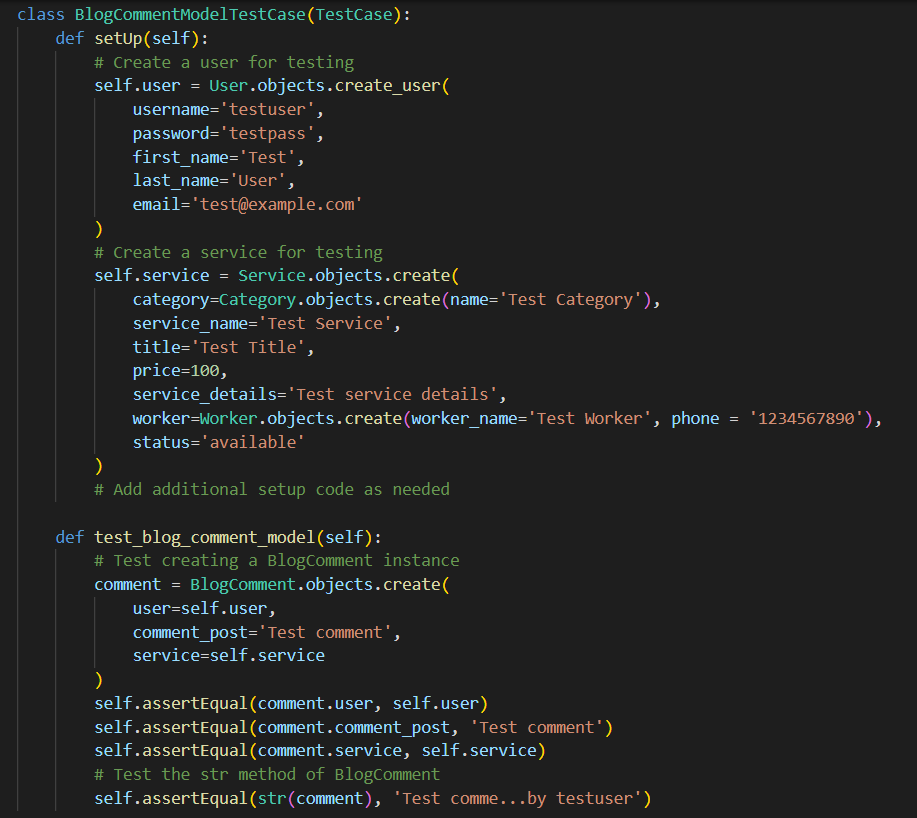


Figure 10: Testcase for BlogComment



Figure 11: Testcase for ShippingAddress

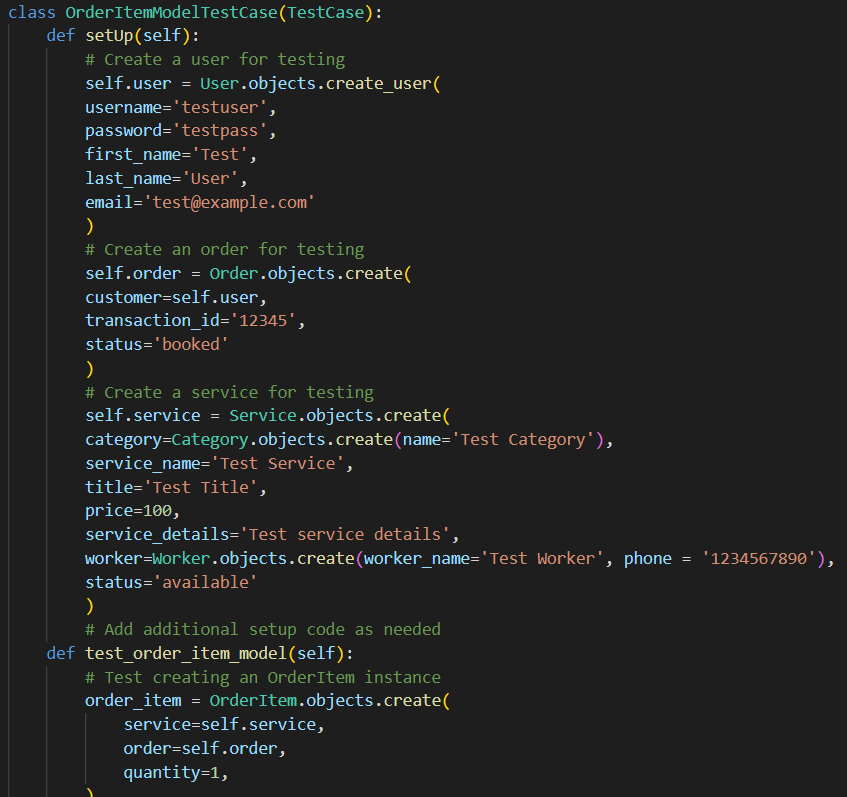


Figure 12: Testcase for OrderItem

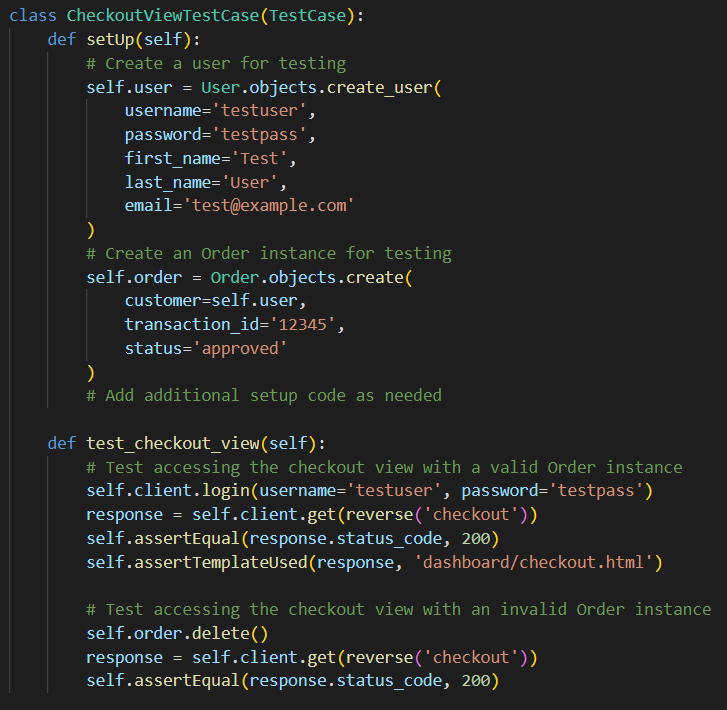


Figure 13: Testcase for Checkout

# Conclusion:

To wind up, the project 'Sipalu' was chosen for the development of a product with an efficient and  
effective home issues solution providing the facility of employment opportunity to the unemployed and sufficient profit to the workers as well as budget friendly to normal customers. As already mentioned above, the project was planned to be performed using Scrum methodology in eight iterations with total team members of 5 people having skills on development and testing along with one being a scrum master with combination and coordination of individual capabilities and working capacity the project was completed. Initially, the user story map of the project was built successfully that expressed and displayed almost all the complete features and functionalities of Sipalu which resulted in creating user story and then the user story details were based on roles as proposed by end user of the system. The system's release plan was developed and successfully executed based on the product and sprint backlogs. This included the total number of working days and the specific release dates for each user story. in such way, the Sipalu web application was completed with the Scrum workflow being based on its values, principles and its artifacts and activities.

# References

* *An overview of the scrum framework* (no date) *Scrum Alliance*. Available at: <https://resources.scrumalliance.org/Article/overview-scrum-framework> (Accessed: January 1, 2023).
* Sacolick, I. (2022) *What is agile methodology? Modern Software Development explained*, *InfoWorld*. InfoWorld. Available at: <https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html> (Accessed: January 1, 2023).
* Chaudhary, A. (2022) *Introduction to git features and commands.*, *Knoldus Blogs*. Available at: <https://blog.knoldus.com/introduction-to-git-features-and-commands/> (Accessed: January 3, 2023).
* Real Python (2021) *Testing in django (part 1) – best practices and examples*, *Real Python*. Real Python. Available at: <https://realpython.com/testing-in-django-part-1-best-practices-and-examples/> (Accessed: January 5, 2023).
* Bigelow, S.J. (2020) *TDD vs. BDD vs. Atdd and other Agile Development Techniques: TechTarget*, *Software Quality*. TechTarget. Available at: <https://www.techtarget.com/searchsoftwarequality/tip/TDD-vs-BDD-vs-ATDD-and-other-Agile-development-techniques> (Accessed: January 5, 2023).
* *A visual guide to version control* (no date) *BetterExplained*. Available at: <https://betterexplained.com/articles/a-visual-guide-to-version-control/> (Accessed: January 5, 2023).
* Aha! (2023) *A guide to user story mapping (with examples)*, *A Guide To User Story Mapping (With Examples) | Aha! software*. Aha! Available at: <https://www.aha.io/roadmapping/guide/release-management/what-is-user-story-mapping#:~:text=User%20story%20mapping%20is%20a,the%20most%20delightful%20user%20experience.&text=By%20visually%20mapping%20out%20these,and%20break%20it%20into%20parts> (Accessed: January 7, 2023).
* (no date) *What is user story?* Available at: <https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/> (Accessed: January 8, 2023).
* Atlassian (no date) *The product backlog: Your ultimate to-do list*, *Atlassian*. Available at: <https://www.atlassian.com/agile/scrum/backlogs> (Accessed: January 8, 2023).
* *Types of software testing* (2022) *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/types-software-testing/> (Accessed: January 8, 2023).