|  |  |  |
| --- | --- | --- |
| **Module** | **Portfolio** | **Assessment Type** |
| Collaborative Development (5CS024) | 1 | Individual Report |

**Appointment Management System-(Frontend-Developer)**

Student Id : 2228160

Student Name : Ritesh Jung Lama

Section : L5CG4

Group : L5CG4 Group 4

Module Leader : Mr. Biraj Dulal

Lecturer : Mr. Biraj Dulal

Submitted on : 02/04/2023

Word Count : 1690

**Acknowledgements**

I have been able to successfully develop the Appointment Management System, thanks to the constant guidance and support of our instructor, Mr. Aadhrsha Khadka. Their valuable inputs and direction have been crucial in making this project a reality, and I express my heartfelt gratitude to them.

While developing this system, I have learned numerous concepts, such as effective time management, communication, and coordination with team members. Additionally, I have gained knowledge on the importance of user experience and the significance of creating a user-friendly interface that is easy to navigate and understand.

I would also like to thank my team members who have played a crucial role in the success of this project. Their support, trust, and commitment to the project have been essential in making this project a success.

Overall, I am proud of the Appointment Management System we have developed, and I am confident that it will be of great use to its intended users.

Table of Contents

[Self-appraisal form 5](#_Toc131353674)

[Personal objectives – performance measurement 5](#_Toc131353675)

[Collaboration Document 7](#_Toc131353676)

[Evidence of good collaboration 7](#_Toc131353677)

[Continuing Personal Development (CPD) 8](#_Toc131353678)

[**Issue tracking** 8](#_Toc131353679)

[Appendix A 9](#_Toc131353680)

[Choosing for the relevant technologies 9](#_Toc131353681)

[Include report 9](#_Toc131353682)

[Implementing functional requirement 10](#_Toc131353683)

[Codes and its explanation 10](#_Toc131353684)

[Use of Version control (Screenshots of logs) 12](#_Toc131353685)

[Appendix 2 15](#_Toc131353686)

[Evidences of Good communication and file sharing 15](#_Toc131353687)

[Evidence of Continuing Personal Development (CPD) 17](#_Toc131353688)

[Evidences of Issue tracking 17](#_Toc131353689)

[Figure 1Main.dart screenshot 11](#_Toc131353658)

[Figure 2 Code folder Structure 12](#_Toc131353659)

[Figure 3 Total commit screenshot 12](#_Toc131353660)

[Figure 4 github commit screenshot 1 13](#_Toc131353661)

[Figure 5 github commit screenshot 2 13](#_Toc131353662)

[Figure 6 github commit screenshot 3 14](#_Toc131353663)

[Figure 7 BaseCamp messages screenshot 1 15](#_Toc131353664)

[Figure 8BaseCamp messages screenshot 2 15](#_Toc131353665)

[Figure 9 email messages screenshot 1 16](#_Toc131353666)

[Figure 10 email messages screenshot 2 16](#_Toc131353667)

[Figure 11 Issue tracking Screenshot 17](#_Toc131353668)

# Self-appraisal form

|  |  |  |  |
| --- | --- | --- | --- |
| Student number | 2228160 | Name | [Ritesh Jung Lama](mailto:np03cs4s220363@heraldcollege.edu.np) |
| Project | Appointment Management System | Date | 02/04/2023 |
| Role | Developer | Team | L5CG4 (group 5) |
| Sprint (1 or 2) | 1 |  |  |

# Personal objectives – performance measurement

|  |  |  |  |
| --- | --- | --- | --- |
| **Objectives** | **Evidence provided** | **Evaluation**  *Student / tutor* | |
| Choosing for the relevant technology | When considering technology for frontend development in mobile app development, Flutter is an excellent tool to create high-performing and scalable applications. Its cross-platform capabilities enable developers to create applications for both Android and iOS with ease, as it eliminates the need for coding in different languages. For writing code, I opted to use Visual Studio Code, a user-friendly platform that is also available for free. | 9 |  |
| *Tutor feedback:* | | | |
| Implementing functional requirement | I have written code for every page in out Mobile  Application (signup,login,booking appointment,  Integrating backed, CRUD Appointment) | 9 |  |

# Collaboration Document

## Evidence of good collaboration

|  |
| --- |
| As a frontend developer, I feel fortunate to be part of a dedicated team that interacts well and shares ideas seamlessly. Our productive meetings, open communication channels, and willingness to help one another have been instrumental in our collaborative efforts. The positive feedback and encouragement that we offer each other motivate us to do better, and our seamless file sharing processes make working together enjoyable. I am proud to be part of such a supportive and enthusiastic team, and I am confident that we will deliver an exceptional frontend experience for our project. |

## Continuing Personal Development (CPD)

|  |
| --- |
| In my role as a Flutter developer, I engaged in comprehensive research to understand the mobile application development process. I employed Google to search for relevant documentation and watched YouTube videos to enhance my understanding of industry best practices. I also conducted research on managing state in Flutter. In instances where I encountered errors that required resolution, I turned to online resources such as Stack Overflow and utilized AI chat assistance to help troubleshoot and resolve the issues. |

## **Issue tracking**

|  |
| --- |
| During the frontend development phase, I uploaded my work onto Git-hub for the team to review and test. Through their testing efforts, they were able to identify minor issues, such as the end time for booking appointments being set prior to the start time. I addressed these issues by implementing necessary validations. Additionally, our project manager suggested enhancements, such as implementing a user avatar in the Sign-Up page and displaying it on the profile page. I took on the task and successfully implemented the requested features. |

# Appendix A

## Choosing for the relevant technologies

### Include report

When it comes to selecting the right technology for frontend development in mobile app development, Flutter emerges as an excellent tool that empowers developers to create highly scalable, robust, and high-performing applications with ease. The greatest advantage of using Flutter is its ability to facilitate cross-platform development, enabling developers to create applications that run seamlessly on both Android and iOS platforms without the need for writing code in different languages. This not only saves developers time and resources but also ensures that the application is consistent in terms of functionality and user experience across different platforms.

From creating clear user interfaces to enhancing app performance, Flutter, in my experience as a mobile app developer, offers a wide range of capabilities and functionalities that expedite the app development process. The development process is more effective and quicker thanks to its hot reload functionality, which enables developers to see the changes they make to the code immediately.

I decided to use Microsoft Studio Code, a robust and user-friendly platform that offers all the required tools and features for developing top-notch applications, to write code in Flutter. Its user-friendly design and sophisticated debugging tools made it simple for me to create and test my app. Additionally, Microsoft Studio Code is free, making it a cost-effective choice for developers trying to create dependable and scalable mobile applications.

# Implementing functional requirement

## Codes and its explanation

In my Flutter app's frontend development, I employed the MVC model and GetX package for state management. The MVC (Model-View-Controller) pattern is widely used for dividing an application into three interconnected components: the model for data and business logic, the view for the user interface, and the controller for handling data flow and view updates. Similarly, GetX is a well-known Flutter state management package that offers a variety of features such as reactive programming, dependency injection, and route management, making it effective for State management.

I opted for the GetX package to manage the state in my Flutter application. GetX is a highly popular package that offers a user-friendly and intuitive approach to managing the application's state. Its capabilities for sharing data and updating the user interface seamlessly across different sections of the app make it an excellent choice. Additionally, with features like dependency injection, reactive programming, and route management, the GetX package provides a comprehensive solution for efficient state management in Flutter applications.

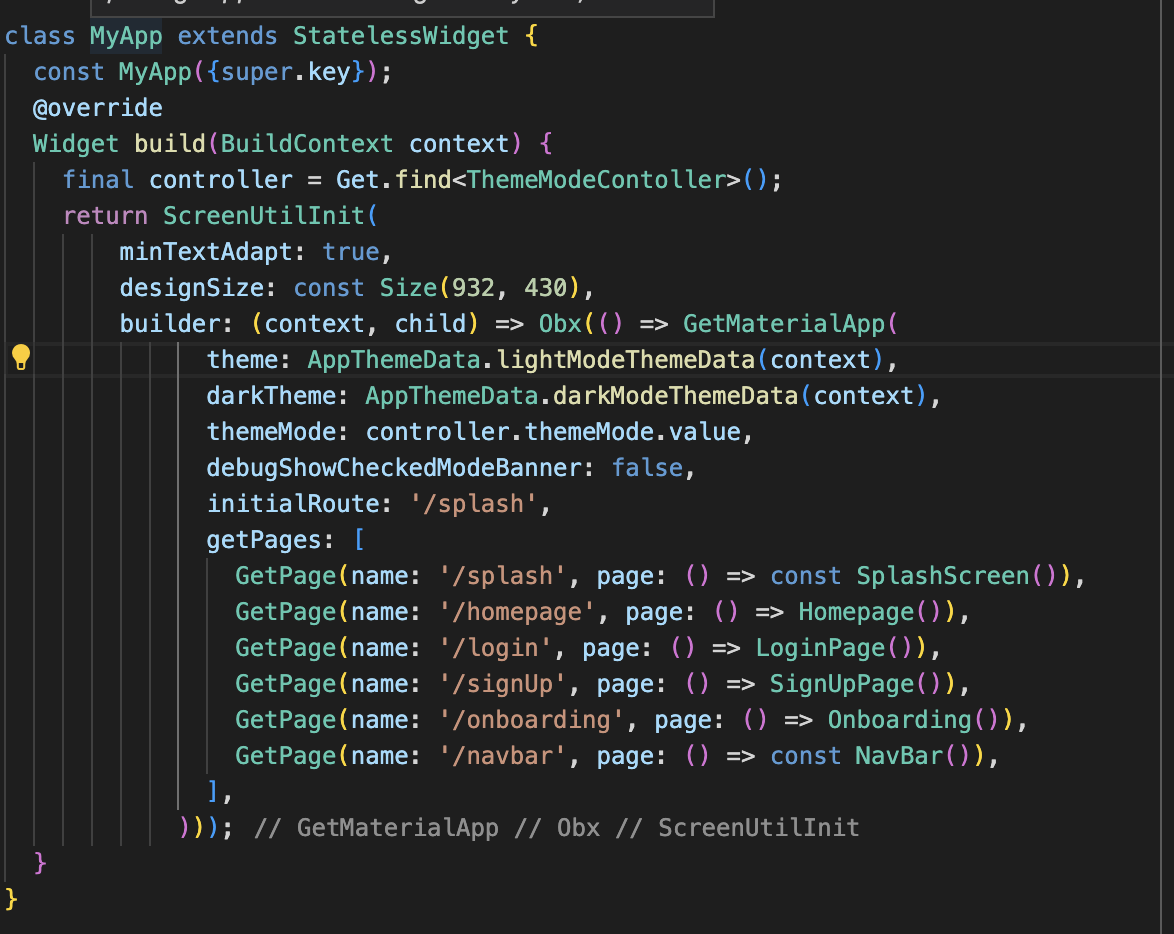


Figure 1Main.dart screenshot

The main code of my program is contained within the MyCode class, which constructs the page and initializes the ScreenUtilInit package. This package is crucial for creating a responsive UI in Flutter and allows for dynamic scaling based on the device's screen size. Once the package has been initialized, the code then returns the GetMaterial page, a page widget that provides additional functionality and design elements to the UI. Overall, utilizing the ScreenUtilInit package and the GetMaterial page widget provides a robust and versatile foundation for building high-quality and responsive UIs in Flutter. Then In the bottom of the code the routes is defines and initial routes is defined which is splash Screen.

**Graphical user interface, application

Description automatically generated**

Figure 2 Code folder Structure

The main code of my program is structured into different folders. The "controller" folder contains the GetX controller, which manages the pages within the code. The "exception" folder contains helper code for Dio Exceptions, which is a popular package used for API handling. The "resources" folder stores the color codes, API endpoints, and base URL. The "view" folder holds all the UI code for the application, while the "services" folder stores the API services code responsible for handling API requests. Finally, the "widget" folder holds reusable widget code that can be used across different parts of the application.

# Use of Version control (Screenshots of logs)

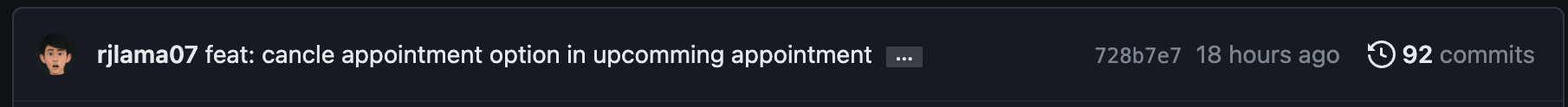


Figure 3 Total commit screenshot

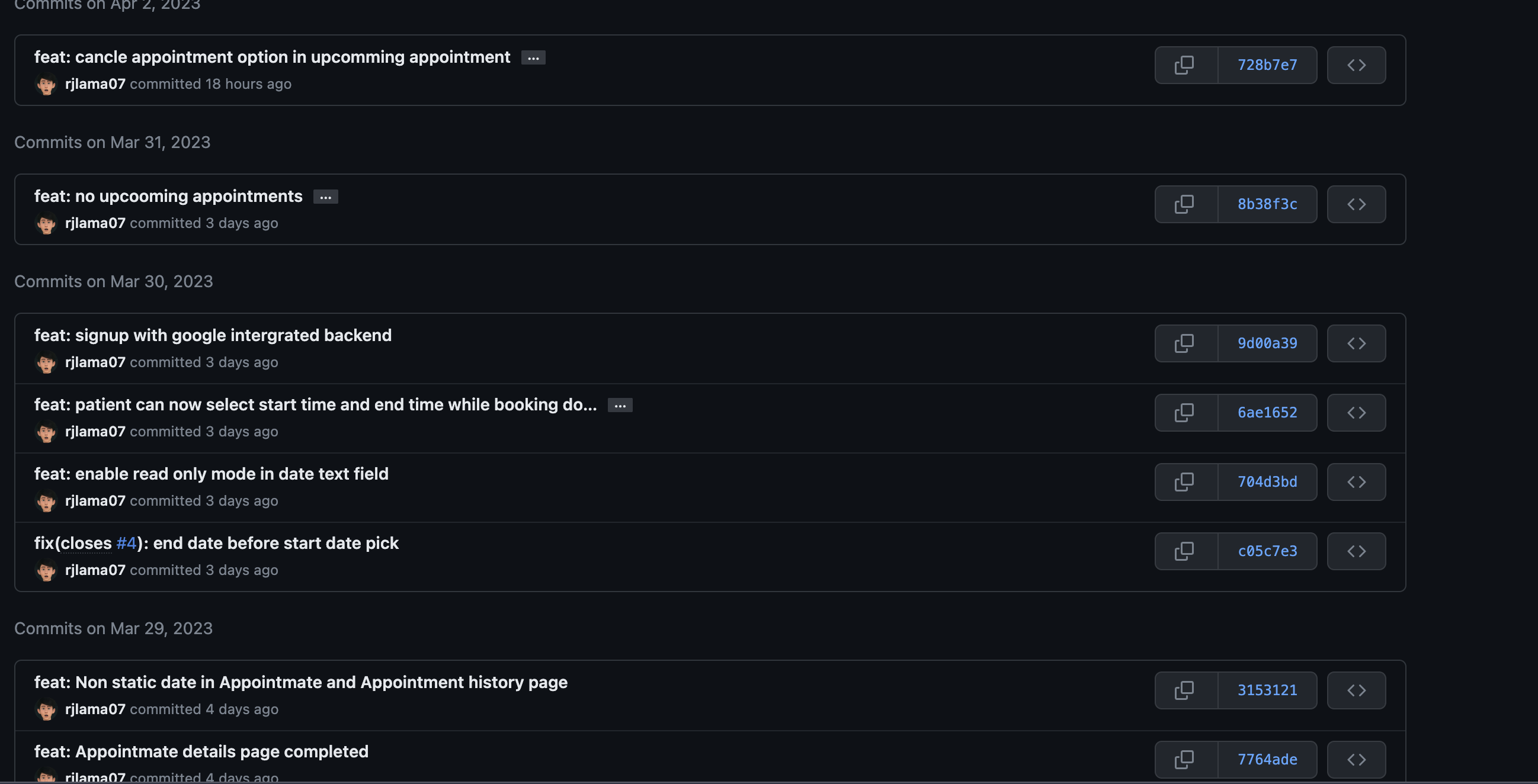


Figure 4 github commit screenshot 1

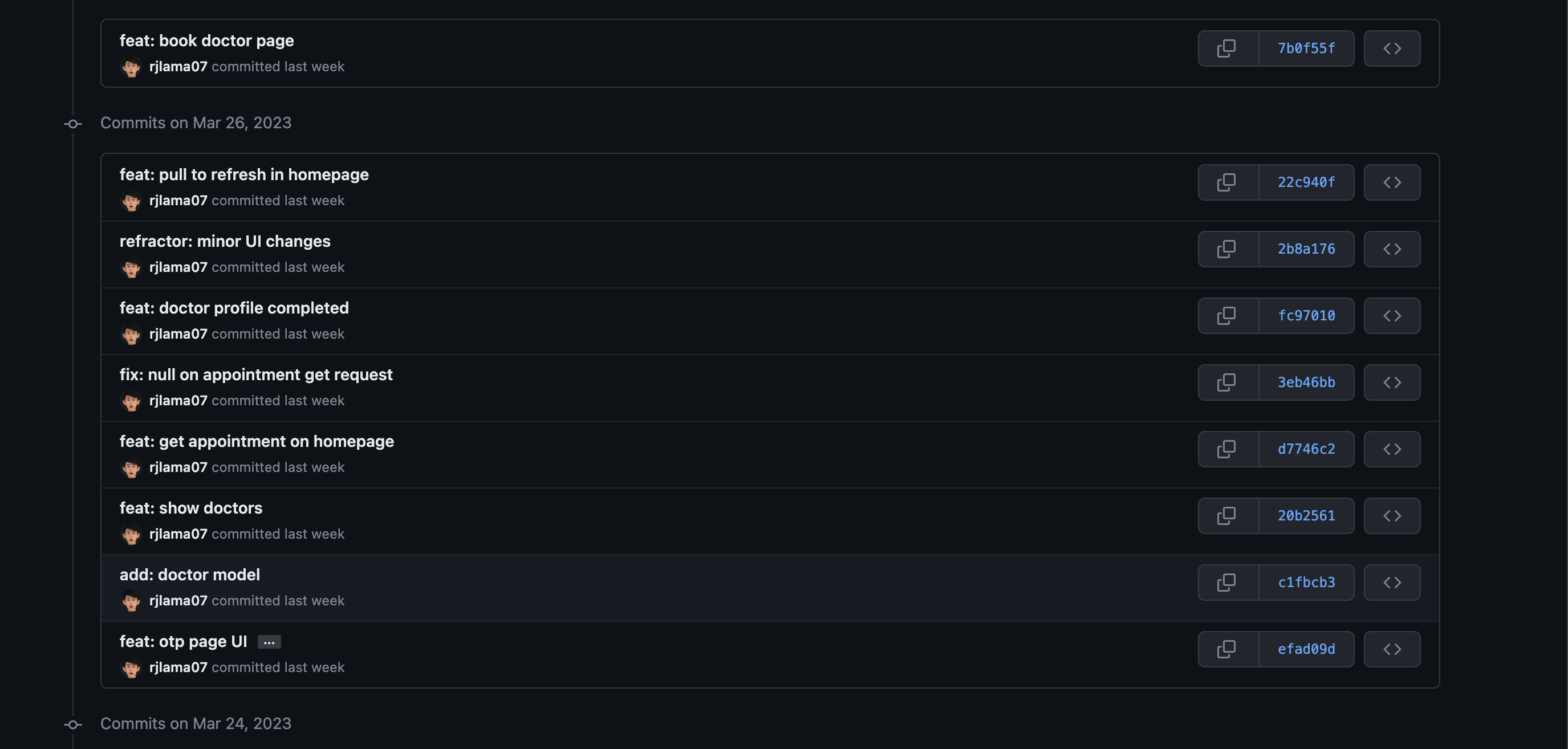


Figure 5 github commit screenshot 2

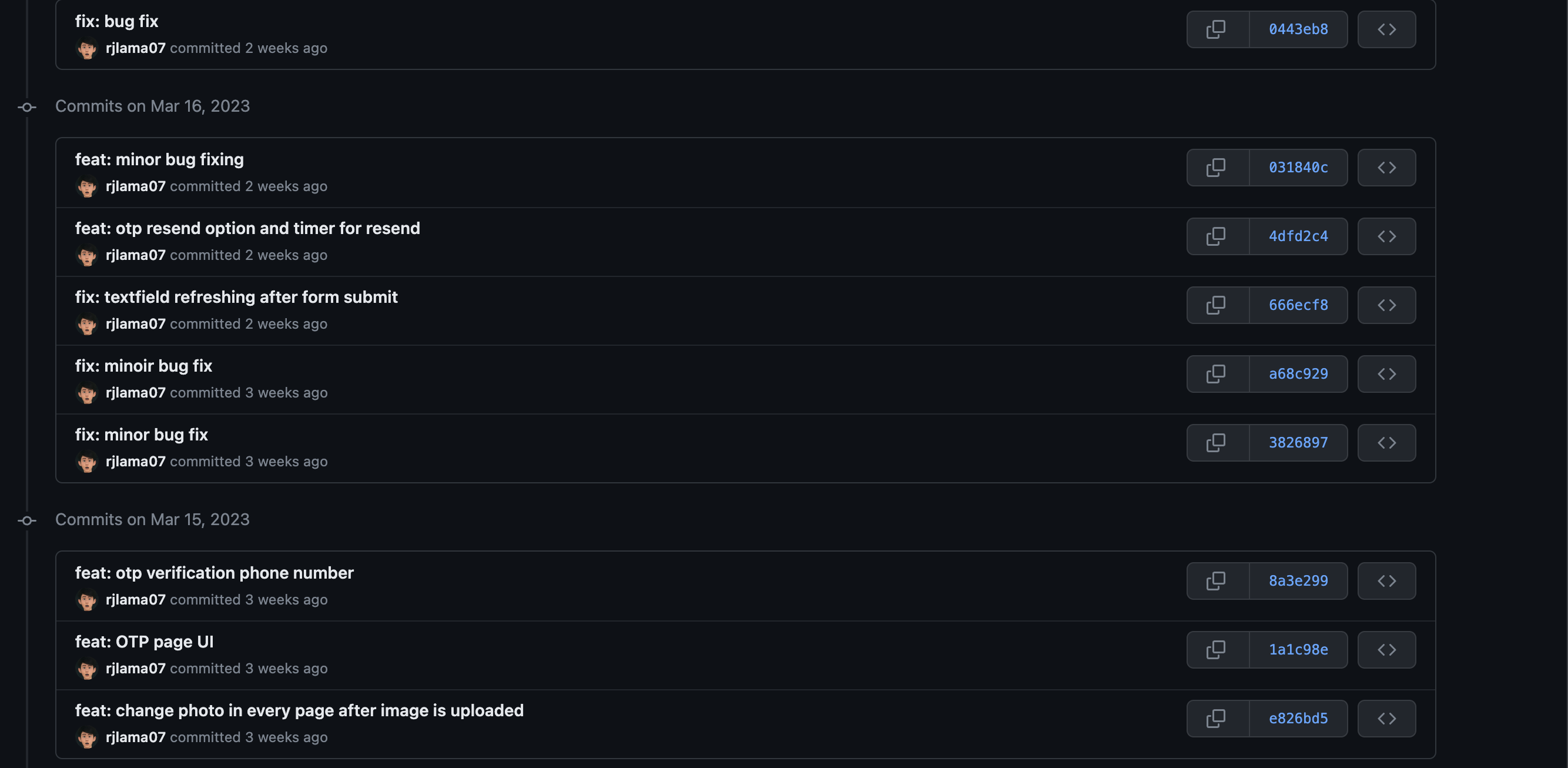


Figure 6 github commit screenshot 3

As a member of our organization's development team, I have been actively contributing to our projects by making numerous commits to our GitHub repository. To ensure effective collaboration and organization, I have made it a priority to follow proper commit message formatting.

Throughout my contributions, I have made a total of 92 commits, reflecting the various changes and improvements made to our codebase. From bug fixes to feature additions, each commit was made with the goal of improving the overall functionality and user experience of our applications.

In addition to making regular commits, I have also taken the time to review and analyze the code changes made by my fellow team members. By doing so, I am able to provide feedback and suggestions for improvement, ultimately leading to a more cohesive and efficient codebase.

Overall, my commitment to maintaining clear and organized commits serves as a testament to my dedication to our organization's development efforts.

# Appendix 2

## Evidences of Good communication and file sharing

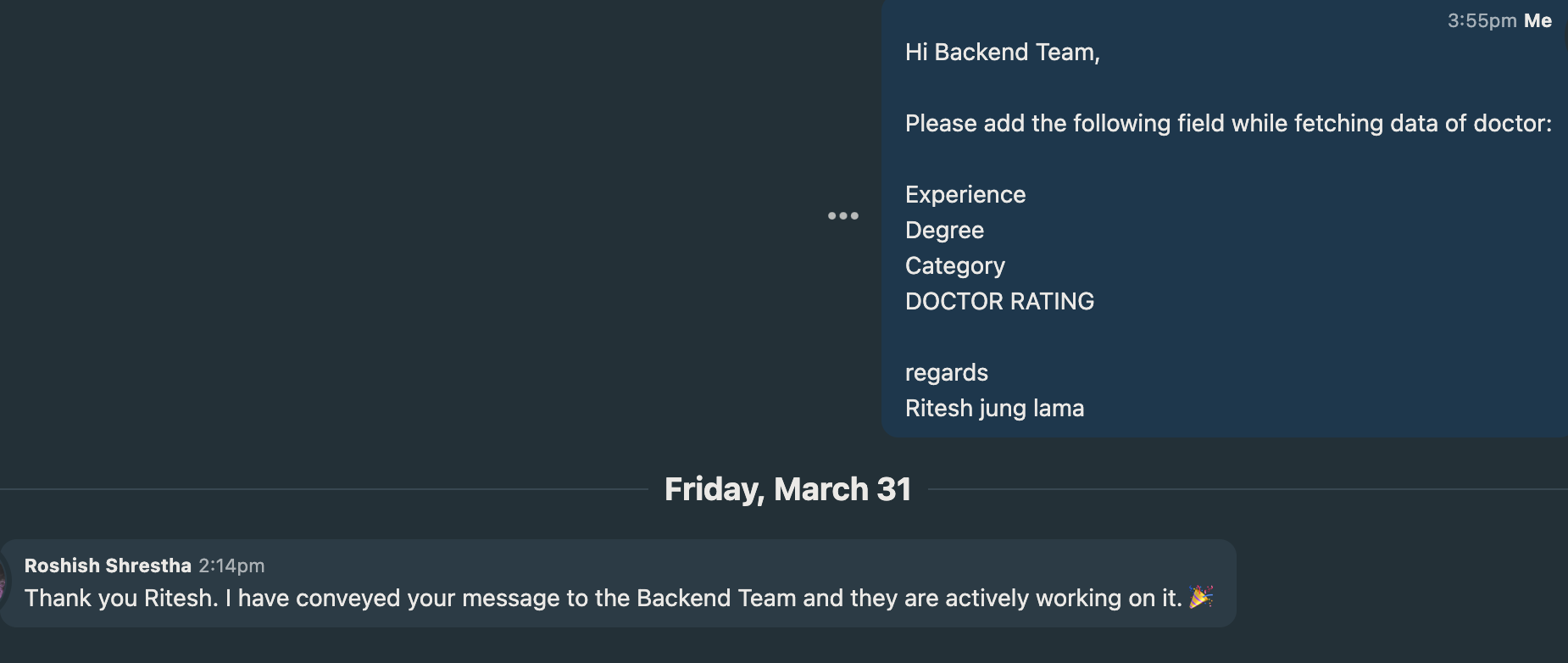


Figure 7 BaseCamp messages screenshot 1



Figure 8BaseCamp messages screenshot 2

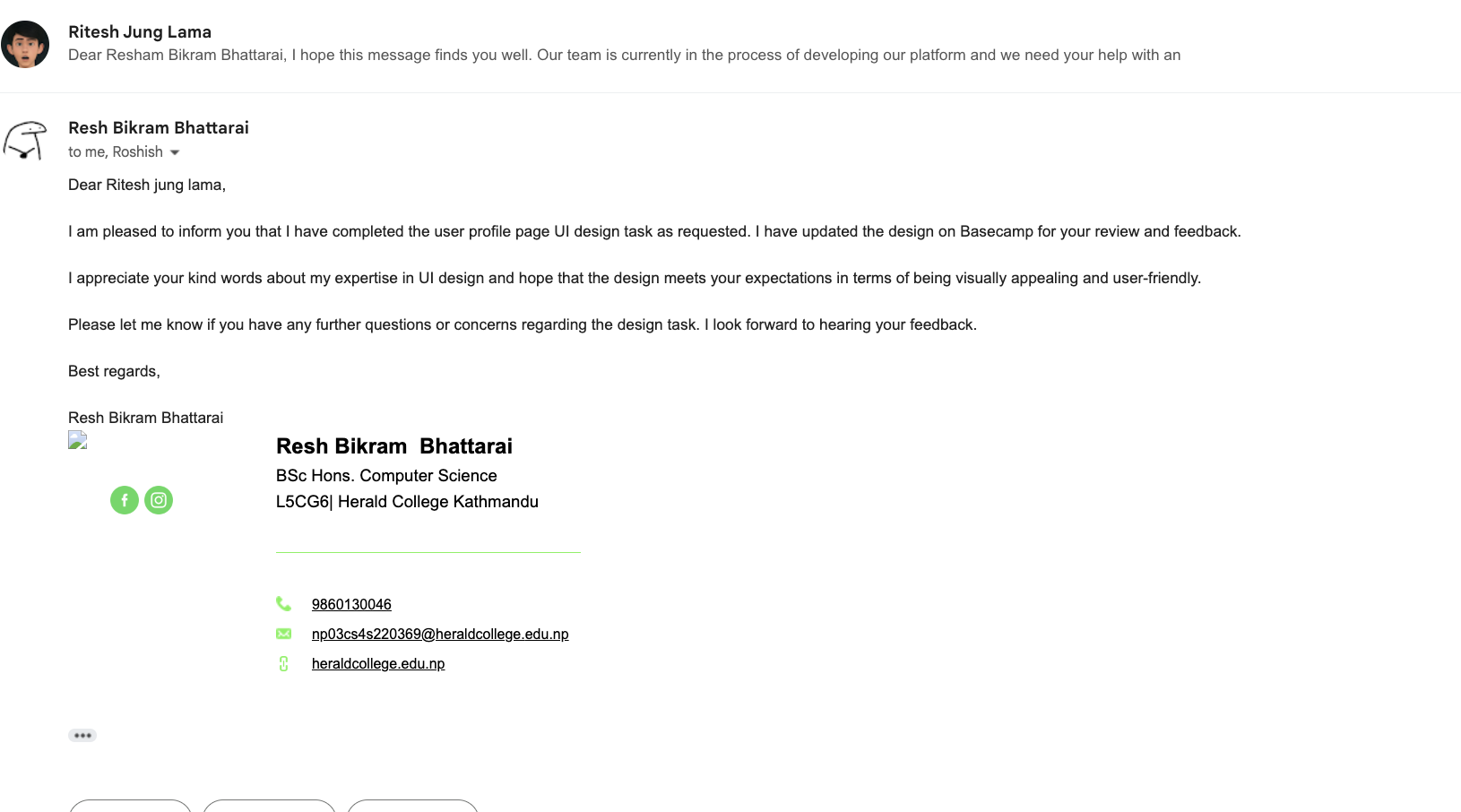


Figure 9 email messages screenshot 1

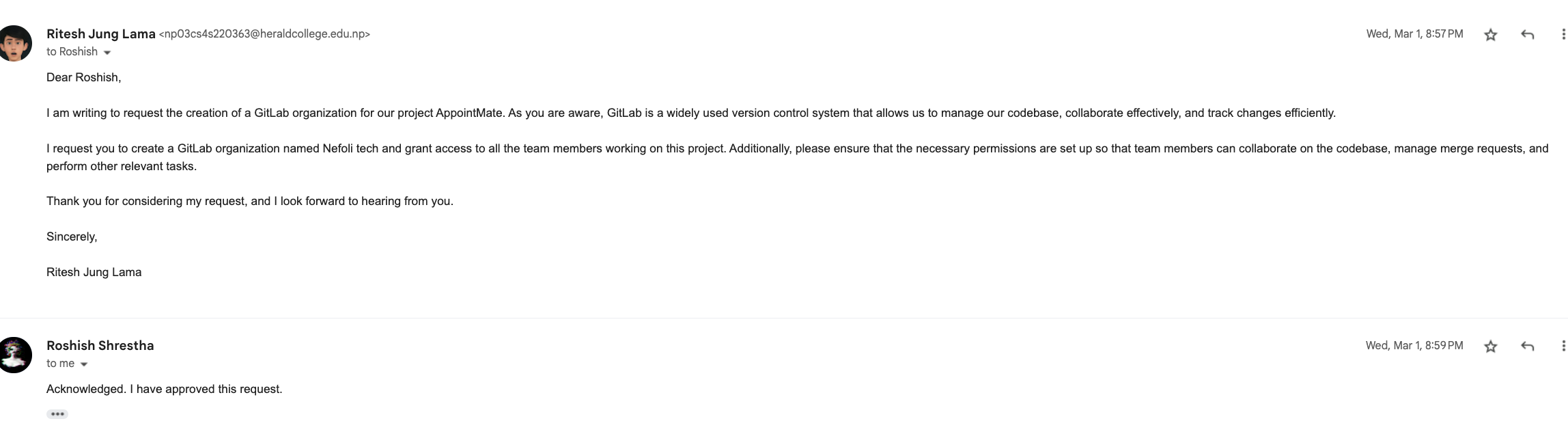


Figure 10 email messages screenshot 2

So these are the screenshots of my message in backend and on need of fast response I have emailed my team mates .

## Evidence of Continuing Personal Development (CPD)

As a dedicated Flutter developer, I recognized the significance of meticulous research in delivering top-notch applications. Consequently, I invested a substantial amount of time in acquiring a deep understanding of the mobile application development process. This involved scouring the internet for relevant documentation, consuming educational videos on YouTube, and perusing industry best practices.

Moreover, I remained up-to-date with the latest advancements in state management in Flutter by following prominent developers in the community on social media and regularly attending virtual meetups and webinars on the subject. This ensured that I had a comprehensive understanding of the latest trends and techniques in the field.

Despite my conscientious approach, I occasionally encountered complex challenges that required expert intervention to overcome. In such cases, I turned to reputable online resources such as Stack Overflow and Quora, which provided invaluable insights and advice from experienced developers. Additionally, I capitalized on the benefits of AI chatbots to troubleshoot and resolve issues expeditiously, allowing me to maintain efficiency and meet project deadlines.

## Evidences of Issue tracking

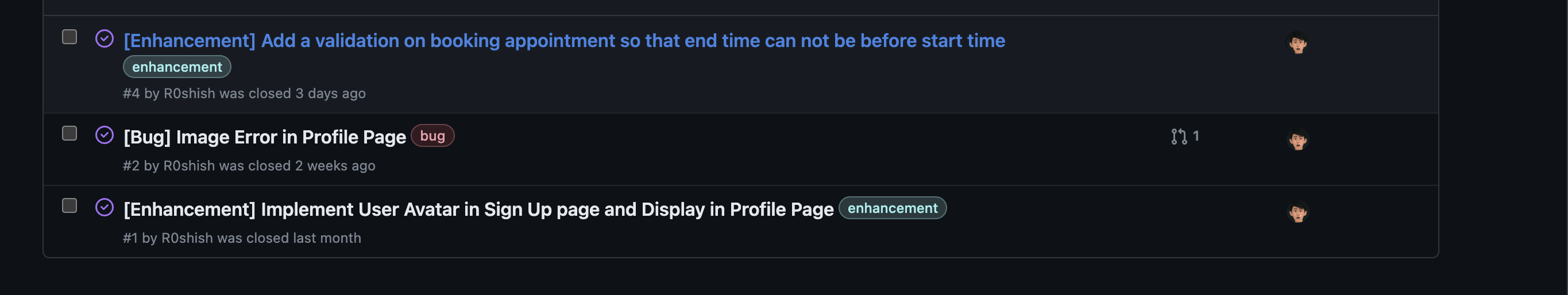


Figure 11 Issue tracking Screenshot

As part of the frontend development phase, I shared my work on Git-hub to enable my team members to review and test it. During their testing, they identified some minor issues, such as the end time for booking appointments being set prior to the start time. I resolved these issues by implementing the necessary validations. Our project manager also suggested some enhancements, including adding a user avatar to the Sign-Up page and displaying it on the profile page. I took on the task and successfully integrated the requested features into the project, resulting in a more user-friendly experience.