A PROJECT REPORT

ON

Online Marketplace

Submitted By:

Atharv Bhardwaj - 'H' - 2115000241

Abhishek Sharma - 'H' - 2115000030

Abhishek Rathore - 'H' - 2115000024

Kunal Gupta - 'H' - 2115000562

In partial fulfillment for the award of the degree of

Bachelor of Engineering

IN

Computer Science

BONAFIDE CERTIFICATE

Certified that this project report **"Online Marketplace"** is the bonafide work of "**Atharv Bhardwaj**, **Abhishek Sharma, Abhishek Rathore and Kunal Gupta**" who carried out the project work under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr.Sandeep Kumar Rathore

(HOD-CSE)

SIGNATURE

SUPERVISOR

Mr.Akash Kumar Chaudhary

(Technical Trainer)

Submitted for the project viva-voce examination held on 15 April 2024.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. We would like to articulate our profound gratitude and to all those people who extended their wholehearted cooperation and have helped us in completing this project successfully.

We are thankful to our mentor Mr.Akash Kumar Chaudhary for teaching and assisting us with the new technology and guiding us at every step, and it wouldn't have been possible for us to finish the project in such short period of time if it were not for his motivation. I would also like to thank all the faculty members who were directly and indirectly contributed in the completion of our project.

CHAPTER 1

INTRODUCTION

1.1. Client Identification/Need Identification/Identification of relevant

The "Online Marketplace" is a user-friendly web based platform designed exclusively for college students, aiming to simplify the process of buying, selling, renting, and lending various items relevant to campus life. This virtual hub provides an accessible platform for students to meet their needs while fostering a sense of community within the college environment.

1.2. Identification of Problem

The “Online Marketplace” project identifies and aims to solve problems related to the inefficiency, lack of community connection, communication challenges, trust issues, accessibility, administrative difficulties, and feedback mechanisms in the current campus trading environment for college students.

1.3. Identification of Task

Develop a user-friendly online platform, the "Online Marketplace," addressing challenges in campus transactions by implementing features such as streamlined item listings, real-time messaging, user profiles, and an administrative dashboard to foster community connection, trust, and efficient exchanges among college students.

1.4. Timeline

Task / Period	Week 1	Week 2	Week 3	Week 4	Week 5
Project Selection					
Mentor Allocation					
Project Planning					
Prototype and Designing					
Documentation					

1.5. Organization of the Report

CHAPTER 2

LITERATURE SURVEY

"Online Marketplace" project involves reviewing existing studies on online platforms for student communities, focusing on user-centric design, real-time messaging integration, and trust-building mechanisms. Explore relevant research on e-commerce platforms, user authentication, and feedback systems to inform the project's methodologies. Investigate successful implementations of Node.js, Express.js, MongoDB, and Socket.io in similar applications. Additionally, examine literature on mobile responsiveness and effective marketing strategies for student-driven platforms.

Significance of CampusMarketplace:

- **Enhancing Campus Efficiency**: The "Online Marketplace" project significantly improves the efficiency of buying, selling, renting, and lending items among college students, replacing traditional methods with a streamlined and centralized online platform.

- **Fostering Community Connection**: The project aims to create a virtual hub that fosters a sense of community within the campus environment, promoting interaction and collaboration among students for their various needs.

- **Trust and Security in Transactions:** By implementing features like user profiles, ratings, and a real-time messaging system, the project addresses trust and security concerns, ensuring a reliable platform for students to engage in transactions with confidence.

- **User-Centric Design and Accessibility**: The focus on user-friendly design and mobile responsiveness ensures that the platform is accessible across various devices, enhancing the overall user experience and making it convenient for students to use the marketplace.

- **Administrative Oversight and Accountability**: The inclusion of an administrative dashboard empowers administrators to manage user accounts, monitor listings, and address reported issues, contributing to the overall integrity and accountability of the platform.

CHAPTER 3

DESIGN FLOW/PROCESS

3.1. Concept Generation

Introduce a dedicated section for virtual campus events within the platform, allowing students to buy, sell, or exchange items related to specific events, fostering a sense of community engagement.

Explore the incorporation of AR features for item previews, enabling users to visualize items in their real-world context before making a transaction, enhancing the online shopping experience.

3.2. Evaluation & Selection of Specifications/Features

The evaluation of features for the CampusMarketplace project was based on the following criteria:

- Select features that align with project objectives, ensuring a secure, scalable, and user-friendly Online Marketplace platform for college students.

- Feasibility of implementation: How technically complex and resource-intensive is it to implement the feature?

3.3. Analysis of Specifications and Features

Evaluate the specifications pertaining to the user interface, emphasizing the need for a user-friendly and intuitive design to enhance the overall accessibility and experience for college students. Assess the specifications related to the integration of technologies (React js, Node.js, MongoDB, Socket.io), ensuring they align with project requirements and constraints, and explore opportunities for optimization or future technology upgrades.

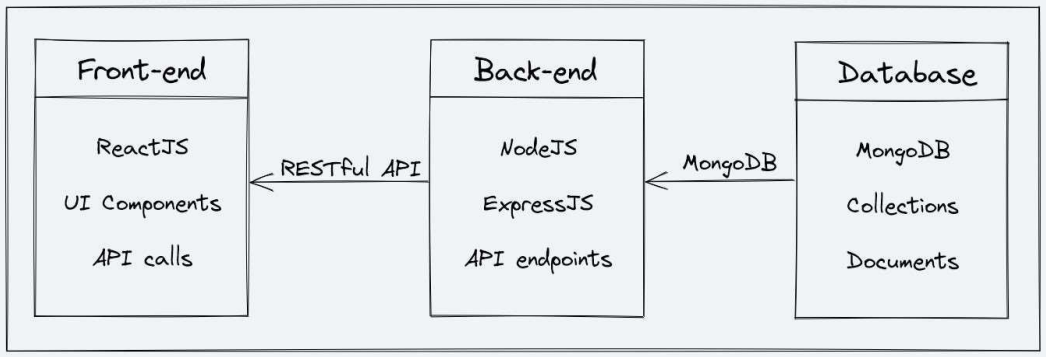
3.4. Design Flow

1. **Homepage** : It has a consistent layout across web and provides Quick access to login or sign up, product details etc.

2. **My Product**: It provides the information/details of selected products .

3. **Add Product**: In this page seller can upload the product.

4. **Favorite**: In this page Buyer can see our selected favorite product or details.



Architecture Diagram

3.5. Implementation Plan

We have implemented the designing of our website with the latest tools available to maintain the compatibility of our website with the modern tech.

Below is the description of the tech stack that we have used in our project.

Technologies used:

Our project is based on front-end and back-end technologies.

3.5.1. For front-end we are using React JS.

> **React JS** is a JavaScript library that is widely used for building user interfaces is one of the most popular front-end development frameworks, and it provides a rich set of tools and features that allow developers to customize the behavior and appearance of their user interfaces. With React,

developers can create unique and engaging user experiences that are tailored to the needs of their users.

React is useful in a project because it allows developers to build complex user interfaces with ease. By using React, developers can create reusable UI components that can be combined to create a variety of different layouts and user experiences. This makes it easier to build *scalable* and *maintainable web applications*.

One of the key features of React is its ability to provide real-time updates to the user interface. By using a technique called "virtual DOM," React can update the user interface without having to reload the entire page. This helps to create a seamless user experience that is *faster* and *more responsive* than traditional web applications.

Another benefit of using React in a project is its ability to work with other JavaScript libraries and frameworks. React is designed to be modular and can be easily integrated with other libraries and frameworks, such as Redux, React Router, and Axios. This allows developers to build powerful and complex web applications that can meet the needs of their users.

React is also useful in a project because it is highly customizable. React provides a rich set of tools and features that allow developers to customize the behavior and appearance of their user interfaces. With React, developers can create unique and engaging user experiences that are tailored to the needs of their users.

In conclusion, React is a powerful and flexible JavaScript library that is useful in a wide range of web development projects. Its ability to provide real-time updates, work with other libraries and frameworks, and provide a high degree of customization makes it an essential tool for building scalable and maintainable web applications. Whether you're building a small website or a large web application, React is a valuable tool that can help you create a great user experience for your audience.

3.5.2 For back-end, Node JS, Express JS, MongoDB and Mongoose are used.

1. **Node JS** is an open-source, cross-platform, back-end JavaScript runtime environment that allows developers to build fast and scalable applications. Node JS interacts with the server to handle the client requests while utilizing the commands that are being designed in Express JS.

   It uses an event-driven, non-blocking I/O model that makes it highly efficient for building real-time applications, especially those that involve large amounts of data.

   It is useful in a project in several ways, here are some of them:

   i. **High performance**: It is built on the V8 JavaScript engine, which is the same engine used by Google Chrome. This makes Node.js *highly efficient* and allows it to handle a large number of requests without affecting the performance of the application.

   ii. **Scalability**: It is highly scalable, which means *it can handle a large number of connections simultaneously without slowing down*. This makes it ideal for building applications that need to handle a large amount of traffic.

   iii. **Single language**: With Node.js, developers can use JavaScript on both the front-end and back-end, making it easy to develop full-stack applications using a single language.

   iv. **Large community**: Node.js has a large and active community of developers who contribute to the development of various libraries and modules. This makes it easy for developers to find solutions to their problems and improve the quality of their applications.

   Node.js is a powerful tool that can help developers build fast, scalable, and efficient applications. Its ease of use and large community make it an attractive choice for developers who want to build full-stack applications using a single language.

2. **Express JS** is a popular open-source web application framework built on top of Node.js. It provides a set of tools and utilities for building web applications and APIs in Node JS. Express JS is known for its simplicity, flexibility, and scalability, and it is widely used by developers to build web applications, APIs, and micro services.

   It provides light-weight framework of Node JS and with it we can perform crud operations and create our own server easily fulfilling our project requirements. It also acts as a *powerful middleware for a range of operations such as authentication, logging and error handling*.

   Some of the key advantages of using Express JS includes:

   i. **Simplicity**: Express JS is a lightweight framework that provides a simple and intuitive API for building web applications and APIs. It has a minimalist design philosophy that emphasizes simplicity and ease-of-use.

   ii. **Flexibility**: It is highly flexible and customizable, allowing developers to build web applications and APIs that meet their specific needs. It provides a modular architecture that allows developers to use only the components they need and replace or extend them as needed.

   iii. **Scalability**: It is highly scalable and can handle high volumes of traffic and requests. It provides a non-blocking I/O model that enables asynchronous processing and supports clustering for horizontal scaling.

   iv. **Middleware**: It provides a powerful middleware system that allows developers to easily add features and functionality to their web applications and APIs. Middleware functions can be used for a wide range of purposes, such as authentication, logging, error handling, and more.

   v. **Routing**: It also provides a powerful routing system that allows developers to easily define the routes for their web applications and APIs. It supports a wide range of HTTP methods, such as GET, POST, PUT, DELETE, and more.

vi. **Integration**: Express.js integrates seamlessly with other popular Node.js libraries and tools, such as MongoDB, Socket.io, and more.

Express.js is a powerful and flexible web application framework that provides a wide range of benefits for developers. It enables developers to build fast, scalable, and customizable web applications and APIs with ease, making it an ideal choice for projects of all sizes and complexities.

3. **MongoDB** is a popular NoSQL document-oriented database that provides a flexible, scalable, and high-performance solution for handling unstructured or semi-structured data. It stores data in JSON-like documents with dynamic schemas, making it easy to store and retrieve complex data structures. Unlike traditional relational databases, MongoDB does not requires predefined tables to store data, which makes it ideal for handling large amount of data.

   Some of the key features of MongoDB are:

   i. **Schema-less Database**: It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection *can hold multiple documents and these documents may consist of the different numbers of fields, content, and size*. It is not necessary that the one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.

   ii. **Document Oriented**: In MongoDB, all *the data stored in the documents* instead of tables like in RDBMS. In these documents, the data is stored in fields (key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

iii. **Indexing**: In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.

iv. **Scalability**: MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.

v. **Replication**: MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

vi. **Aggregation**: It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e. aggregation pipeline, map-reduce function, and single-purpose aggregation methods

vii. **High Performance**: The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

4. **Mongoose** is an Object Data Modeling (ODM) library for MongoDB. It defines a strongly-typed-schema, with default values and schema validations which are later mapped to a MongoDB document. It provides an incredible amount of functionality around creating and working with schemas. Mongoose currently contains eight Schema Types that a property is saved as when it is persisted to MongoDB.

CHAPTER 4

RESULT ANALYSIS AND VALIDATION

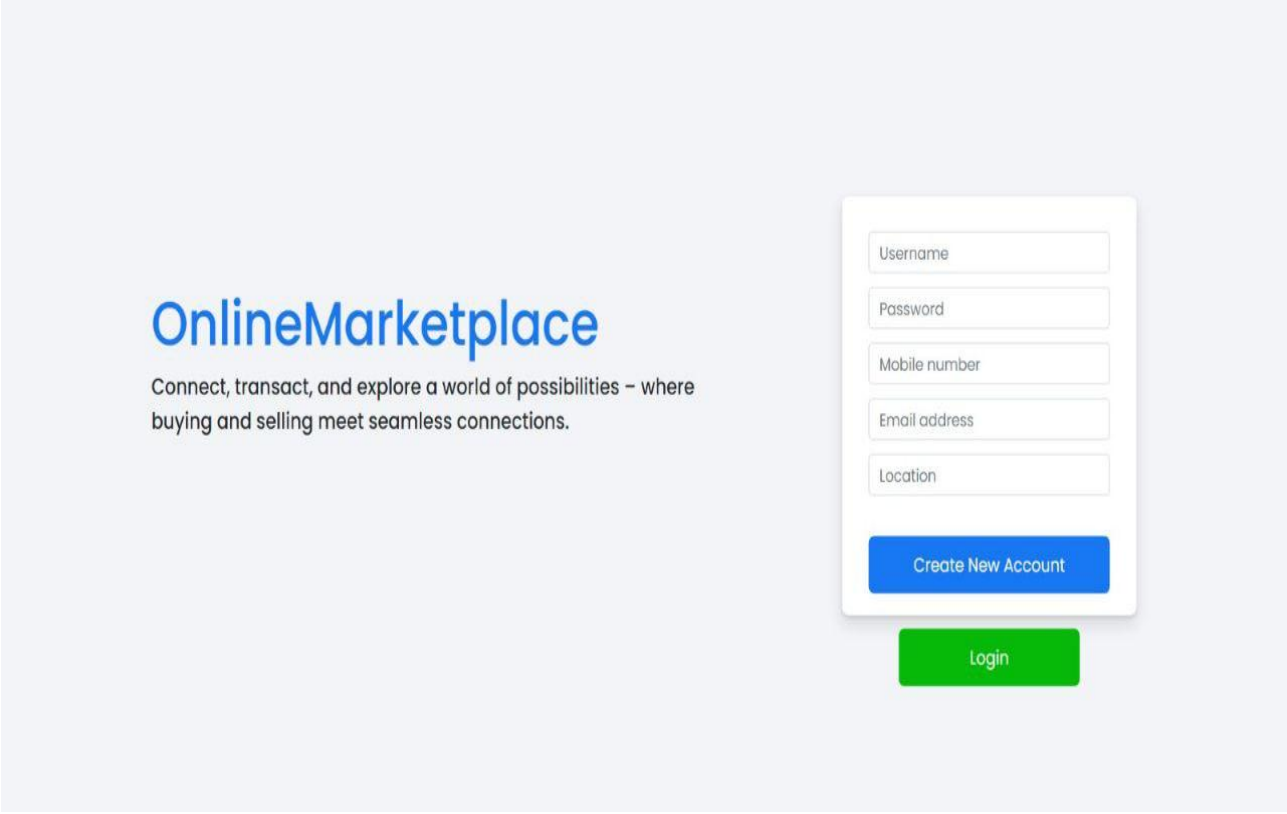


Figure 4.1 : Sign up / Login

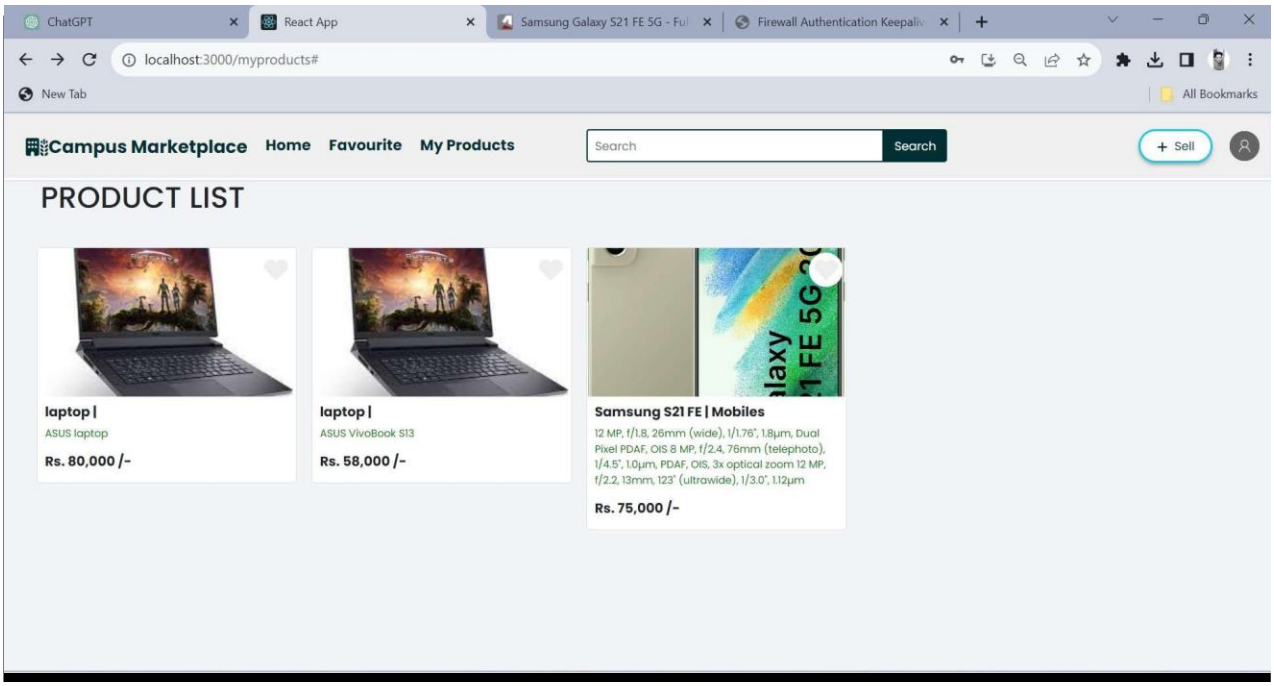


Figure 4.2: Home Section

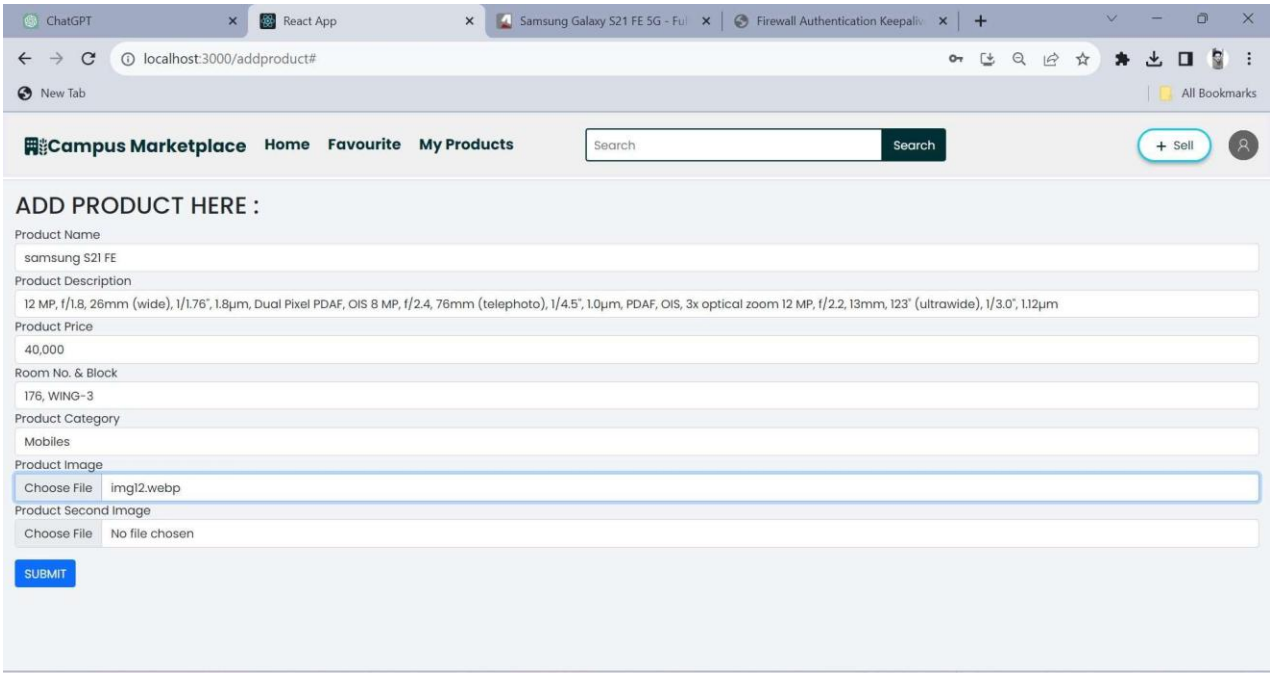


Figure 4.3: Add Product Section

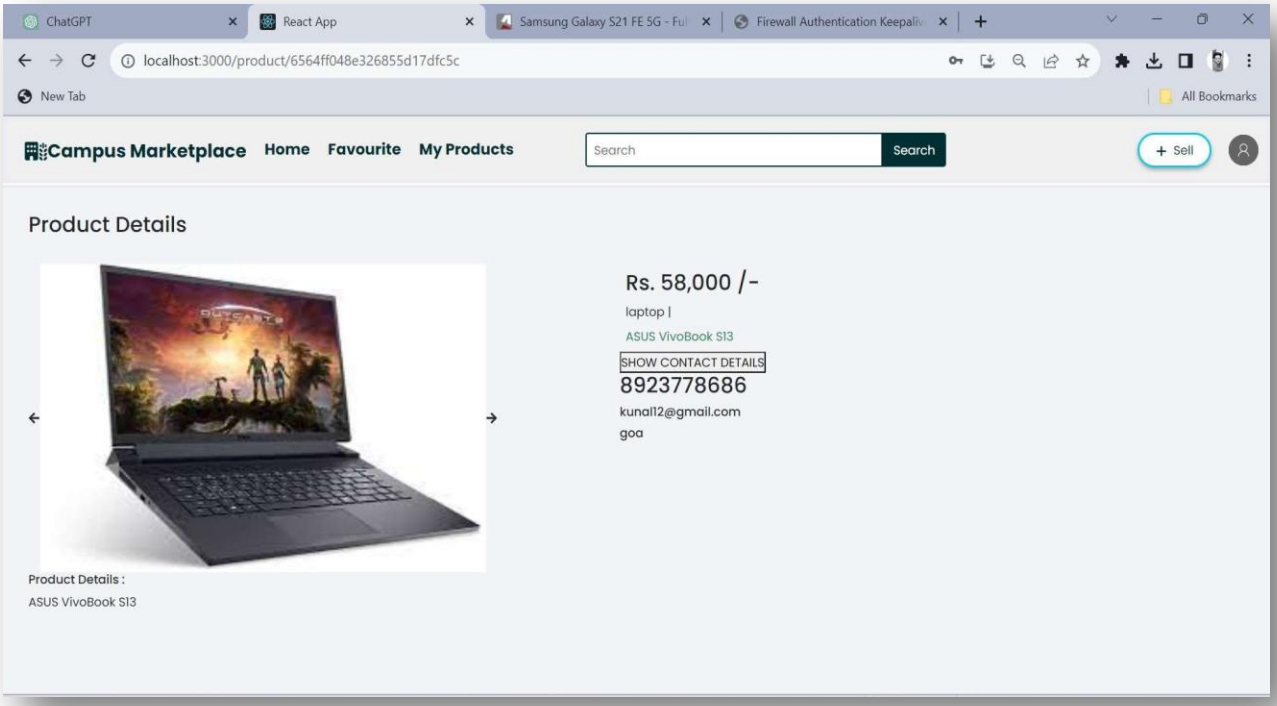


Figure 4.4 Product Details Section

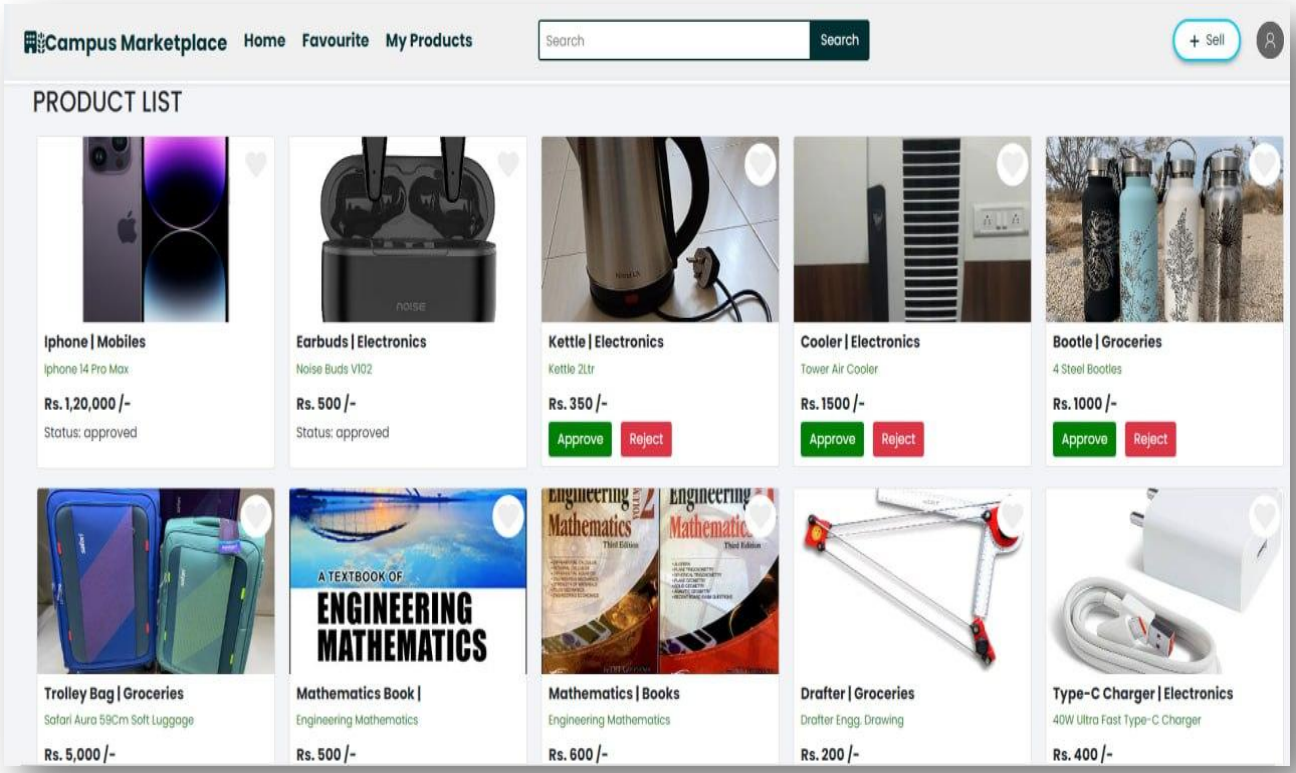


Figure 4.5: Admin Approval Page

CHAPTER 5

CONCLUSION AND FUTURE WORK

6.1. CONCLUSION

The Online Marketplace project seeks to create a convenient platform for college students to exchange items, fostering a sense of community and sustainability on campus. By leveraging modern web technologies and a user-centered design, the project aims to demonstrate the positive impact technology can have on student life within academic communities.

6.2. FUTURE WORK

As the platform continues to evolve, future work should focus on enhancing its capabilities and expanding its reach.

Integration of AI and Machine Learning: Exploring the integration of AI and machine learning algorithms could enhance the platform's ability to understand user behavior, leading to more accurate item recommendations and improved search functionality.

Block chain Technology for Security: Researching and incorporating block chain technology could provide an additional layer of security and transparency in transactions, further addressing trust concerns and ensuring the integrity of the marketplace.

Integration with Campus Services: Exploring partnerships or integrations with campus services, such as student organizations or campus events, could broaden the platform's scope, making it a central hub for various aspects of campus life beyond buying and selling.

REFERENCES

- (https://www.w3schools.com/REACT/DEFAULT.ASP)
- (https://cloud.mongodb.com/v2/650ef0bf22e73c75d4f03f###/overview)
- MDN Documentation