"Plotify : An Online Plot Booking System"

A Major Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Engineering in Computer Science Engineering

Submitted by:

Aniruddha Singh Chawda (0827CS211020)
Adarsh Underiya (0827CS211005)
Aarush Yadav (0827CS211003)
Aniruddha Sharma (0827CS2223D05)
Aman Patel (0827CS2223D07)

Guided by:

Prof. Shraddha Sharma
Professor , CSE

Acropolis Institute of Technology & Research, Indore
Aug - Nov 2024

EXAMINER APPROVAL

The Minor Project entitled ***"Plotify : An Online Plot Booking System"*** submitted by **Aniruddha Singh Chawda (0827CS211020), Aniruddha Sharma (0827CS223D05), Aman Patel (0827CS223D07), Adarsh Underiya (0827CS211005), Aarush Yadav (0827CS211003)**has been examined and is hereby approved towards partial fulfilment for the award of ***Bachelor of Technology degree in Computer Science Engineering*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project only for the purpose for which ithas been submitted.

(Internal Examiner) **(External Examiner)**

Date **Date:**

RECOMMENDATION

This is to certify that the work embodied in this minor project entitled **"*Plotify : An Online Plot Booking System*"** submitted by **Aniruddha Singh Chawda (0827CS211020), Aniruddha Sharma (0827CS223D05), Aman Patel (0827CS223D07), Adarsh Underiya (0827CS211005), Aarush Yadav (0827CS211003)**is a satisfactory account of the bonafide work done under the supervision of ***Dr. Kamal Kumar Sethi***, is recommended towards partial fulfilment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

.

(Project Guide)

(Project Coordinator)

(Dean Academics)

STUDENTS UNDERTAKING

This is to certify that the minor project entitled "Plotify : An Online Plot Booking System" has developed by us under the supervision of Dr. Kamal Kumar Sethi. The whole responsibility of the work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Aniruddha Singh Chawda(0827CS211020),
Adarsh Underiya (0827CS211005),
Aarush Yadav (0827CS211004),
Aniruddha Sharma (0827CS223D05),
Aman Patel (0827CS223D07)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are a number of people without whom this project's work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Asst. Prof. Shraddha Sharma ,** AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, Professor & Head CSE, AITR Indore for his support, suggestion, and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

We are grateful to **our parents** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say, “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Aniruddha Singh Chawda(0827CS211020),

Adarsh Underiya (0827CS211005),

Aarush Yadav (0827CS211004),

Aniruddha Sharma (0827CS223D05),

Aman Patel (0827CS223D07)

Executive Summary

"Plotify : An Online Plot Booking System"

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of ***Asst. Prof. Shraddha Sharma***.

This Plot Booking System utilizes cutting-edge web development technologies to offer an efficient and intuitive platform for booking and managing land plots. With our comprehensive interface, users can seamlessly browse and reserve plots based on specific criteria, while administrators benefit from automated processes that enhance operational efficiency.

Key words: Plot Booking System, Efficient Platform, Real-Time Updates, Online platform,

List of Figures

Table of Contents

Chapter 1. Introduction

The Plot Booking System is a web-based application designed to streamline the process of booking and managing land plots. Utilizing advanced web development technologies, this system aims to simplify user interactions, making it easy for customers to browse, filter, and reserve plots based on their preferences. The intuitive interface not only enhances the user experience but also ensures that administrative tasks such as tracking bookings, managing records, and generating reports are automated and efficient. This project addresses the complexities involved in land plot management by providing a secure and comprehensive solution tailored for both end users and administrators.

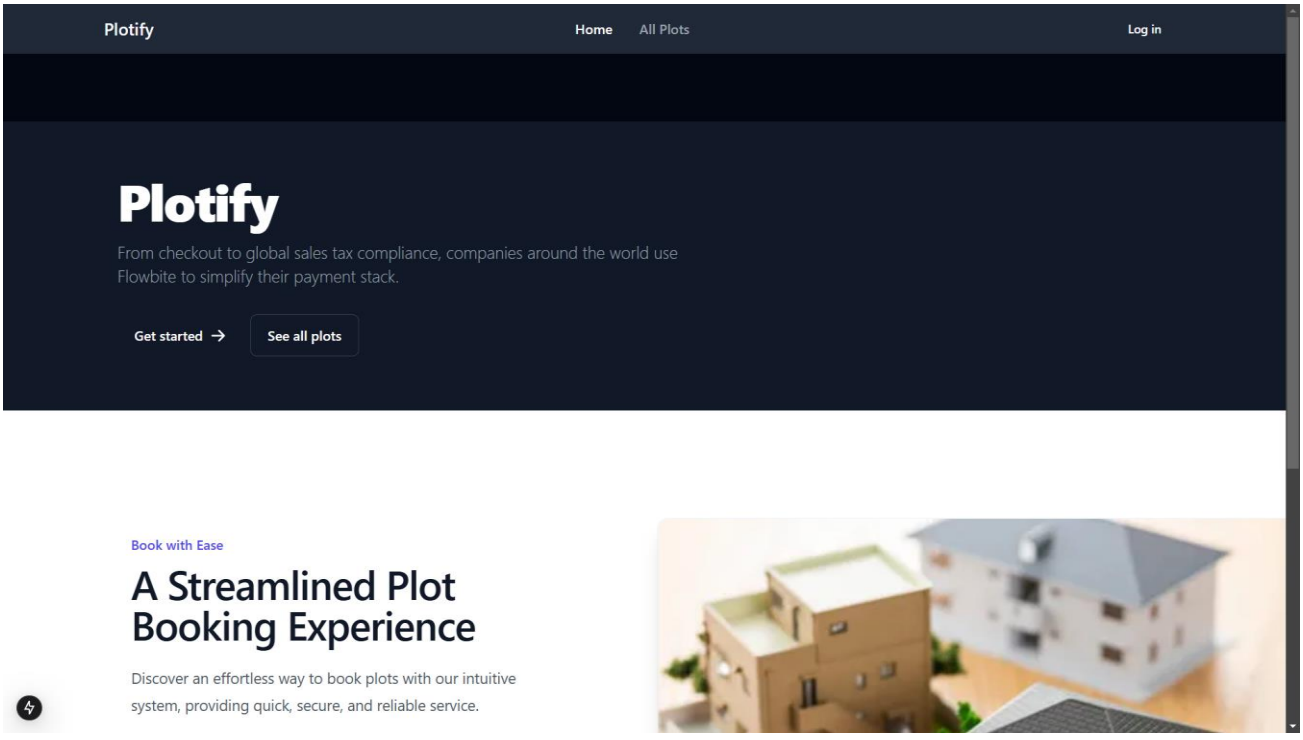


Figure 1.1 Home Page of Plotify Platform

This platform not only provides an intuitive user experience but also enhances administrative control by automating repetitive tasks and reducing manual errors. It is crafted to support scalability, enabling it to accommodate growing datasets and a large number of users. This Plot Booking System is ideal for real estate agencies, developers, and individual users looking for a simple yet powerful way to manage plot reservations and sales.

By offering a secure environment for transactions and reliable data protection, the system ensures user information remains safe, fostering trust and confidence among users.

The Plotify Platform is a groundbreaking initiative tailored for those who want to book and rent their plots . With a mission to simplify the land plot booking and management process, we harness the power of modern web technologies. Utilizing the MERN stack along with Tailwind CSS for a seamless and responsive user experience, our Plot Booking System reimagines real estate operations. The platform is designed to empower users by providing an intuitive and efficient way to reserve and manage plots while streamlining administrative tasks for enhanced productivity.

1.1 Overview

The Plot Booking System is a sophisticated digital platform thoughtfully developed to cater to the intricate demands of property seekers and real estate administrators. It transcends traditional property management by creating a dynamic and efficient space where land booking is simplified and streamlined. The platform is built on cutting-edge technology, incorporating the MERN stack for seamless user experience and Tailwind CSS for a modern, responsive design, while ensuring data integrity and security through well-structured backend frameworks.

It goes beyond conventional property booking by offering personalized features tailored to the unique needs of each user. From intuitive plot search and reservation to comprehensive management tools, the system empowers users to make informed decisions and track their investments effectively, while administrators enjoy automated workflows and improved operational control.

The Plot Booking System aims to revolutionize the real estate landscape by creating a dynamic environment where property transactions are easy to navigate, challenges are minimized, and efficient management becomes a reality.

In essence, the Plot Booking System delivers an unparalleled property management experience, blending innovation with functionality. By leveraging state-of-the-art technology and focusing on user empowerment, the platform transforms the way plots are booked and managed, paving the way for a future where real estate operations are intuitive, transparent, and streamlined.

1.2 Background and Motivation

Imagine a scenario where individuals searching for land plots or managing real estate face numerous challenges, from inefficient reservation processes to the lack of transparent and organized booking systems. That's the gap the "Plot Booking System" project aims to bridge. It's about empowering users with an intuitive and reliable digital platform that transforms how plots are discovered, reserved, and managed.

The motivation behind the Plot Booking System stems from a desire to simplify real estate operations and ensure that both property seekers and administrators have an efficient and user-friendly experience. Our project addresses the common issues of manual booking processes, delayed transactions, and the complexities of managing land data. We want to make it easier for users to find their ideal plot, streamline the booking process, and give administrators the tools needed for efficient oversight.

By utilizing advanced technologies like the Next.Js , Supabase and Tailwind CSS, our platform provides a modern, interactive, and secure environment for plot booking and management. Our vision goes beyond just simplifying real estate tasks; it’s about offering a comprehensive solution that enhances user experience, reduces operational hurdles, and creates a reliable space for all stakeholders involved.

Through the Plot Booking System, we aim to redefine the real estate booking experience. By integrating innovative technology and user-centric features, we aspire to build a supportive ecosystem where property management is seamless, transparent, and effortlessly accessible to all.

1.3 Problem Statement and Objectives

The real estate industry, particularly the process of booking and managing land plots, often faces significant challenges. These include inefficient manual processes, a lack of real-time availability updates, and insufficient user-friendly interfaces that hinder both property seekers and administrators. Users struggle with navigating outdated systems, while administrators experience difficulties in efficiently managing reservations and keeping data organized. These challenges necessitate a modern solution that streamlines plot booking, offers transparency, and simplifies real estate management for all parties involved.

Thus, the system implemented has the following objectives:

1. To develop a comprehensive and user-friendly interface where users can browse and book available plots effortlessly based on specified criteria.

2. To create an administrative portal that simplifies the management of plot data, automates booking confirmations, and provides insights for better operational efficiency.

3. To implement features that enhance the user experience, such as real-time updates on plot availability, search filters, and secure payment processing, ensuring a smooth and efficient booking journey for all users.

1.4 Scope of the Project

The scope of the Plot Booking System project is to create an integrated and user-friendly platform that facilitates seamless plot reservation and management. The system will feature an intuitive interface for users to browse, filter, and book plots based on specific criteria, ensuring a smooth and efficient experience. For administrators, the platform will offer a comprehensive dashboard to manage plot listings, track bookings, and generate reports for improved decision-making. The system will incorporate automated notifications, payment integration for secure transactions, and a robust database to manage plot details and availability. Additionally, the platform will provide analytics to help administrators monitor performance and user engagement. Scalability and security are also central to the project, ensuring the system can handle growing user demands while protecting sensitive data. By providing an all-encompassing solution for both users and

administrators, the Plot Booking System will streamline the plot booking process, making it more efficient, transparent, and secure.

1.5 Team Organization

Aniruddha Singh Chawda:

Focused on backend development, ensuring smooth application operations. My tasks included implementing critical functionalities and providing valuable insights during the implementation process. Additionally, I actively participated in testing procedures, offering essential feedback for continuous improvement.

Adarsh Underiya:

Coordinated project planning and designed the user interface, focusing on user experience. Additionally, I managed project documentation and reporting, ensuring accurate progress tracking and successful project delivery.

Aarush Yadav:

Led the deployment process, ensuring a seamless transition of our application into the live environment. My responsibilities included managing collaborative planning discussions, contributing valuable insights to project decisions, and actively participating in various project planning aspects.

Aniruddha Sharma:

Played a fundamental role in our project, focusing on technical aspects. I designed the system's robust architecture and implemented essential backend functionalities .

Aman Patel:

Played a fundamental role in our project, focusing on technical aspects. I designed the system's robust architecture and implemented essential testing functionalities , ensuring the core operations were efficient.

1.6 Report Structure

The project ***Plotify : An Online Plot Booking System*** is primarily concerned with the ***An E-Booking*** and the whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope, and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- This chapter will describe about the related work of the other research to gain more understanding of the project idea. The concept of graphical password will be described in this chapter. The existing conventional password and the benefits of the graphical password authentication will be discussed in this chapter from the reading material and sources such as articles, journals, related websites and existing project.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. Review of Literature

The Plot Booking System project integrates the latest web technologies to offer a robust solution for managing and booking land plots online. By utilizing the MERN stack (MongoDB, Express.js, React, and Node.js), the system provides a highly scalable and responsive platform for both users and administrators. Research in real estate technology demonstrates the growing demand for digital solutions that simplify property transactions, making platforms like the Plot Booking System crucial in modernizing the industry. The system's focus on seamless user experience aligns with studies in human-computer interaction, which emphasize the importance of intuitive design in online platforms. By providing easy-to-use interfaces, secure payment gateways, and real-time availability tracking, the system enhances user satisfaction and boosts engagement.

2.1 Preliminary Investigation

2.1.1 Current System

- Plotify , powered by the Next.js , Supabase and Tailwind CSS, revolutionizes booking platforms by offering real-time interaction capabilities.
- Ongoing research endeavors to enhance Plotify's security and user experience, focusing on innovative encryption methods and intuitive interfaces for privacy and user satisfaction.

Some current system of E-Booking platform in today's world are:

Magicbrickes

Magicbricks is a leading online real estate platform in India that connects buyers, sellers, and renters of residential and commercial properties. It provides a wide range of services including property listings, search filters, and virtual property tours, enabling users to make informed decisions. Magicbricks also offers tools for property valuation, legal advice, and home loan assistance, making it a comprehensive solution for real estate transactions. Its user-friendly interface and

extensive database make it one of the most popular platforms for property buying, selling, and renting in India.

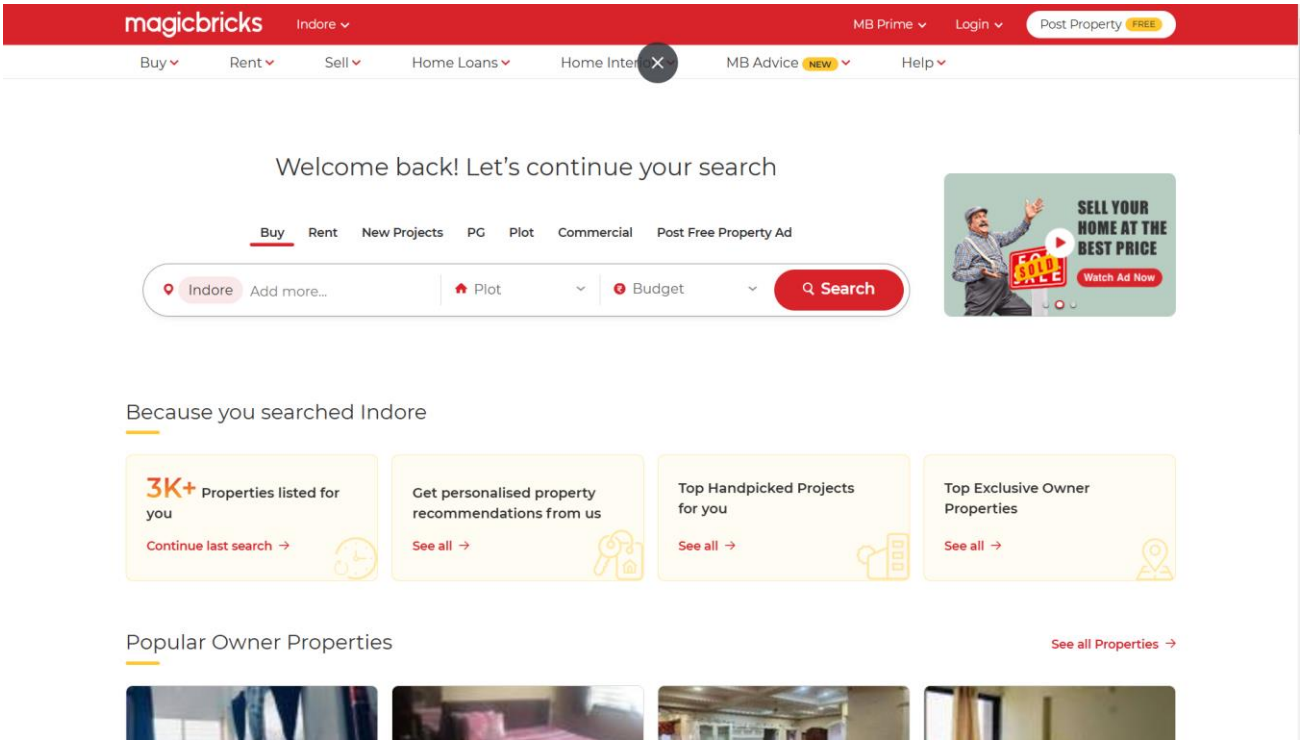


Figure 2.1 UI of MagicBricks (Referred from: Indiatoday.in)

NoBroker

*NoBroker is an innovative real estate platform that eliminates the need for brokers in property transactions. By connecting property owners directly with potential tenants or buyers, NoBroker helps users save on brokerage fees. The platform offers a variety of services, including property listings, home rental, property buying and selling, and even home services like packers and movers. With features such as verified listings, detailed property information, and an easy-to-use interface, NoBroker simplifies the real estate process, making it more transparent and cost-effective for users. It's particularly popular in India for its commitment to offering a hassle-free experience without intermediary agents.

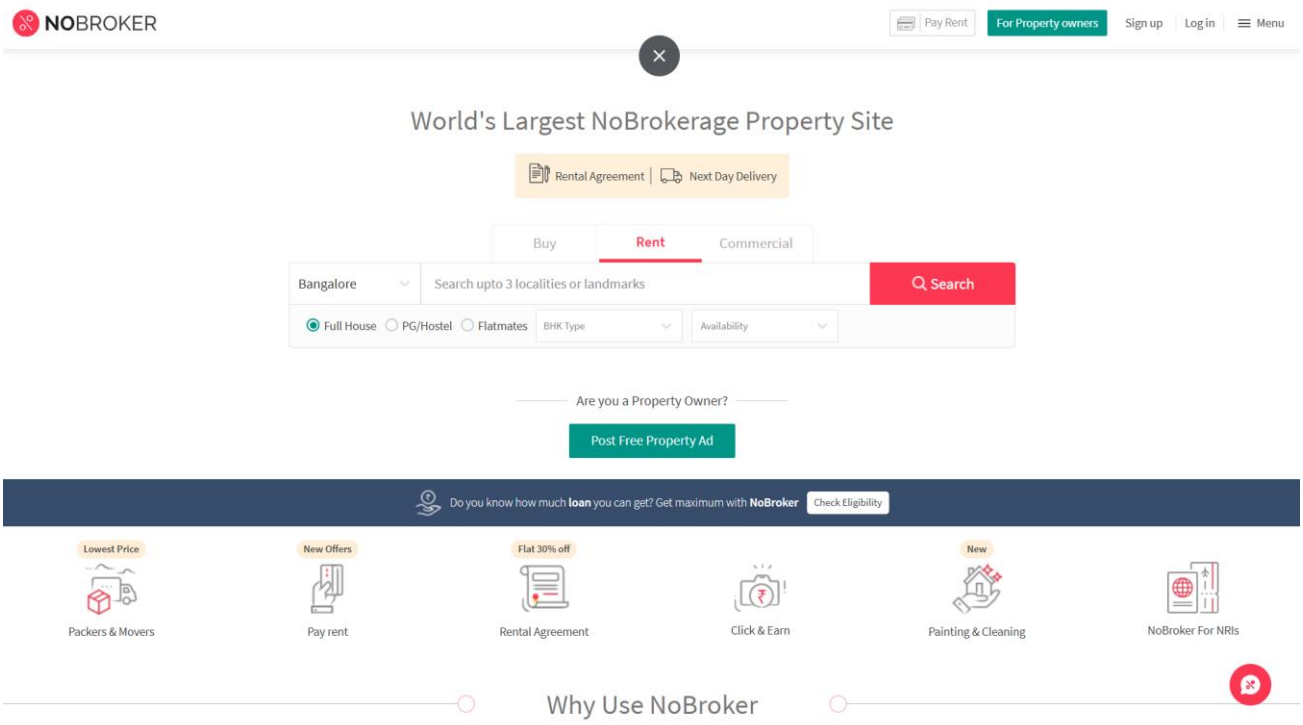


Figure 2.2 UI of NoBroker (Referred from: Lifewire.com)

2.2 Limitations of Current System

The limitations of these are as follows:

- **Lack of Personalization:** Although Magicbricks offers advanced search filters, the system still lacks deep personalization features that could predict and recommend properties based on user behavior and preferences.
- **Inaccurate Listings:** Some property listings may be outdated or inaccurate, leading to potential frustration for users.
- **Limited Support for Property Owners Selling Their Homes:** Property owners trying to sell their homes directly may face challenges in reaching a broad audience compared to traditional brokers who often have larger networks.
- **No In-person Assistance:** Since the platform is digital, users might find the lack of face-to-face interaction or personalized service a limitation, especially in complex transactions.

2.3 Requirement Identification and Analysis for Project

The **EducationTimeline** project requires a thorough process of requirement identification and analysis to ensure its success. Functional requirements include a secure user registration and authentication system, personalized user profiles for both mentors and mentees, an intelligent mentor-mentee matching algorithm, and real-time messaging capabilities for seamless communication. Additionally, the platform should offer a comprehensive repository of study materials, resources, and mock tests, an intuitive scheduling system for booking mentorship sessions, and a dynamic search functionality to find relevant mentors and resources. Non-functional requirements focus on ensuring optimal performance with fast load times and smooth interaction, scalability to handle growing user numbers, and high reliability to guarantee minimal downtime. Security measures, such as encryption and compliance with privacy standards, are vital to protect sensitive user data, while a user-friendly interface, cross-platform compatibility, and adherence to accessibility standards will ensure an inclusive experience for all users. Continuous performance monitoring, load testing, and regular software updates will be essential for maintaining system performance and addressing user needs. Finally, the platform will offer robust support and maintenance features to

assist users and ensure long-term success. These combined requirements form the foundation for a platform that effectively facilitates mentorship, prioritizing user satisfaction, security, and accessibility.

2.3.1 Conclusion

Identifying and analyzing the requirements of **EducationTimeline** ensures that the platform meets the needs of both mentees and mentors while delivering an exceptional user experience. The functional and non-functional requirements together lay a solid foundation for the successful development of a dynamic, secure, and scalable mentorship platform.

Chapter 3. Proposed System

3.1 The Proposal

The Online Plot Booking System is a web-based platform designed to automate and streamline the process of purchasing and reserving real estate plots. The system aims to bridge the gap between real estate developers and potential buyers by offering a seamless online experience for plot searching, booking, and payment. This proposal outlines the core functionalities and benefits of the system and how it will address the existing challenges in plot booking processes.

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

- **Enhanced Customer Experience:** The platform will provide users with a hassle-free experience when searching for and booking plots. Users can view all available plots, their prices, and details from the comfort of their homes, without needing to visit multiple real estate offices.
- **Increased Efficiency for Real Estate Developers :** Real estate companies will benefit from an automated system that reduces the administrative burden of managing plot bookings.
- **Time and Cost Saving :** Both buyers and developers will save significant time and costs associated with manual booking processes. Buyers no longer need to visit the real estate office multiple times to inquire about available plots or finalize bookings.
- **Transparency and Trust :** Both buyers and developers will save significant time and costs associated with manual booking processes. Buyers no longer need to visit the real estate office multiple times to inquire about available plots or finalize bookings.

3.3 Feasibility Study

A feasibility study is conducted to assess the practicality of a project in terms of technical, economic, operational, legal, and schedule feasibility. This study ensures that the project is viable before full-scale development begins. For the Online Plot Booking System, the feasibility study evaluates the likelihood of successfully implementing the system and its potential benefits and challenges.

3.3.1 Technical

Availability of Technology: The required technologies, such as web development frameworks, databases, and payment gateway APIs, are widely available and well-documented.Technologies like HTML5, CSS, JavaScript, and backend frameworks (Node.js, Python Django/Flask, or PHP) are readily accessible for building the frontend and backend of the system. Databases like MySQL or MongoDB can efficiently handle user, plot, and transaction data.

3.3.2 Operational

The use of secure communication protocols (SSL, HTTPS) and encrypted databases ensures that technical risks related to security and data privacy are mitigated.

3.4 Design Analysis

3.4.1 Data Flow Diagrams

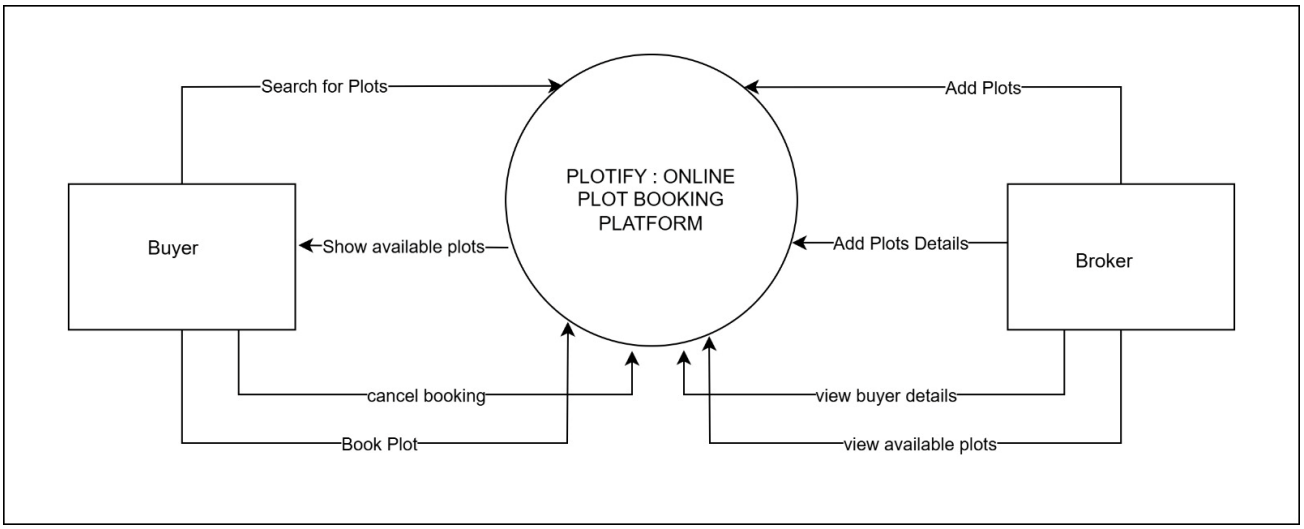
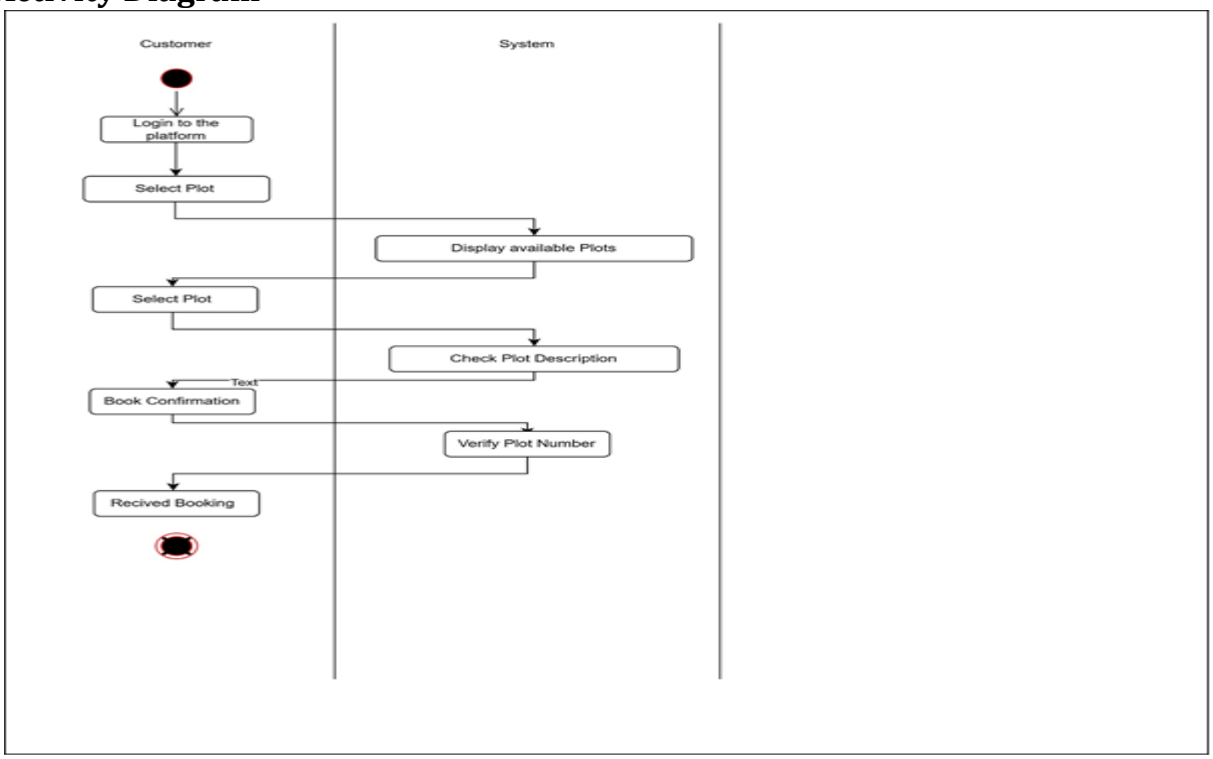


Figure 3.4.1 Level 0 DFD of Plotify Platform*

3.4.2 Design Representation

Activity Diagram



3.5.4 Archicture Diagram

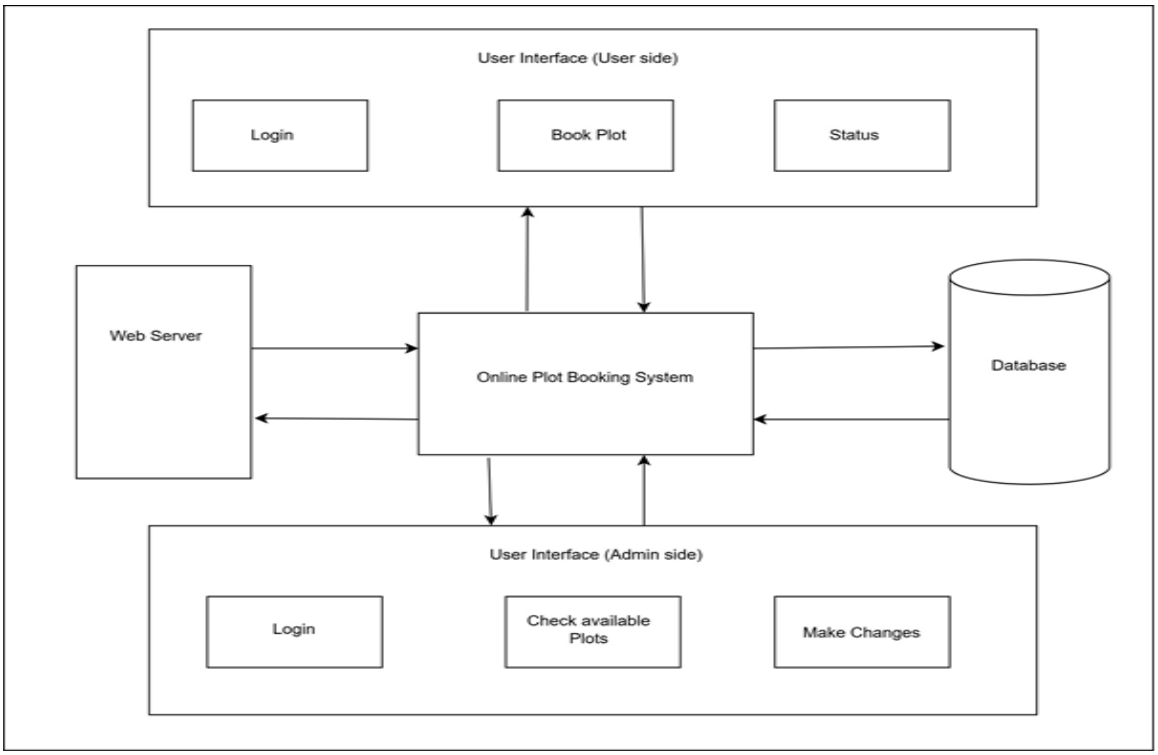


Figure 3.5.4 Acrhitecture Diagram of Plotify Platform

3.5.5 Database Structure

The name of the database created is “db_detect” and there is one table in the database named “logs” for storing the records.

The “Logs” table has the following structure :

Name	Data Type	Description
Datetime	Timestamp	Shows the complete date and time when the person/vehicle enters and is identified
Type	Varchar2	Displays the type of object for example Person, Car, Dog.
CIF	Number	Count per frame.It tells the number of objects in frame.

Table 2 : Database Structure

3.5 Deployment Requirements

There are minimal requirements (hardware, software and services) required to successfully use this system. These are mentioned below:

3.5.1 Hardware

- 64-bit operating system, x64-based processor
- Windows 11 or later operating system
- CPU: Intel Pentium 6300 MHz (minimum)
- RAM: 8.00 GB (7.34 GB usable)

3.5.2 Software

- Node.js – For Frontend
- Tailwind CSS – For Styling
- Supabase – For backend & database management.
- Windows - 7/8/10/11

Chapter 4. Implementation

Implementing an online plot booking system can be broken down into phases that cover key functional and technical aspects. Here's a step-by-step guide to implementing this project, including technologies you might use for each phase.

4.1 Requirements Analysis & Design

4.1.1 Functional Requirements:

- User registration and login.
- Search and filter plots by various criteria (e.g., location, size).
- Book a plot, including payment and booking confirmation.
- Admin panel for managing plots, bookings, and users.

4.1.2 Non-Functional Requirements:

- Scalability for handling multiple bookings simultaneously.
- Data security, especially for payment transactions.
- A responsive, user-friendly interface.

4.1.3 System Design:

- Entity-Relationship Diagram (ERD) for the database.
- Use Case Diagrams to represent major interactions.
- Sequence Diagrams (as described previously) for process flows.
- Well-suited for large and complex projects where frequent reassessment and adaptation are necessary.

4.1.4 Scrum Framework:

- Scrum is an agile framework that organizes development work into fixed-length iterations called sprints. It promotes collaboration, adaptability, and continuous improvement.
- Involves regular sprint planning, daily stand-ups, and sprint reviews.

4.1.5 DevOps Practices:

DevOps emphasizes collaboration and communication between development and operations teams. It incorporates continuous integration, continuous delivery, and automation for faster and more reliable software deployment.

4.1.6 User Acceptance Testing (UAT):

- UAT involves end-users testing the system to ensure it meets their requirements and expectations.
- Users provide feedback on functionality, usability, and any issues encountered during testing .

4.2 Technology used

4.1.1 MERN Stack

Plotify leverages the MERN stack – MongoDB, Express.js, React, and Node.js – to craft a robust mentoring platform. MongoDB serves as the database, efficiently storing user profiles and mentorship data. Express.js facilitates smooth communication between the frontend and backend, handling server-side logic and API endpoints. React powers the frontend, offering a dynamic and intuitive user interface for mentorship connections. Node.js provides the runtime environment, ensuring high-performance server operations. Together, these technologies empower Plotify with speed, flexibility, and real-time functionality, delivering a seamless mentoring experience for users.

Frontend:

Node:

In Plotify, Node.js makes our website look awesome and work smoothly. It's like building with blocks - we create different parts of the site, like buttons or profiles, and put them together easily. Node helps our site load super fast by only changing the parts that need updating, making it feel snappy. Plus, it keeps track of everything happening on the page, so you see new messages or updates instantly without refreshing. With Node, Plotify becomes more than just a website; it's a place where mentors and mentees connect effortlessly for valuable learning experiences.

Backend

Supabase:

Supabase is like the powerhouse behind Plotify. It runs our website's server and handles all the heavy lifting behind the scenes. Just like a traffic cop, it directs the flow of data between users and our database, ensuring everything runs smoothly. Supabase is lightning-fast and efficient, perfect for handling lots of users chatting and connecting at once. Plus, it's super flexible, allowing us to easily add new features and scale up as our community grows. With Supabase ,Plotify stays responsive and reliable, giving users a seamless experience every time they log in.

MongoDB:

EducationTimeline relies on MongoDB as its database backbone. MongoDB is like a digital filing system, organizing user data efficiently. Its flexibility allows EducationTimeline to store various types of information, from user profiles to chat messages. With MongoDB's real-time synchronization, EducationTimeline ensures seamless communication among users, enhancing the overall experience.

Tailwind CSS:

Tailwind CSS empowers EducationTimeline with its utility-first approach, streamlining the design process. With Tailwind, EducationTimeline can rapidly create responsive and visually appealing user interfaces. Tailwind's extensive set of pre-built components and utilities simplify styling, enabling consistent and customizable designs across the platform. Its flexibility and ease of use make Tailwind CSS an ideal choice for EducationTimeline's frontend development.

4.2 Tools Used

4.2.1 VS Code

Visual Studio Code (famously known as VS Code) is a free open-source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

4.2.2 Draw.io

Designed by Seibert Media, draw.io is proprietary software for making diagrams and charts. The software lets you choose from an automatic layout function or create a custom layout. They have a large selection of shapes and hundreds of visual elements to make your diagram or chart one-of-a-kind. The drag-and-drop feature makes it simple to create a great looking diagram or chart.Draw.io has options for storing saved charts in the cloud, on a server, or network storage at a data center, depending on your needs.

4.3 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

Here's a testing plan broken down by types of testing:

1. Unit Testing

   Write unit tests for individual components, such as API endpoints, UI components, and utility functions.

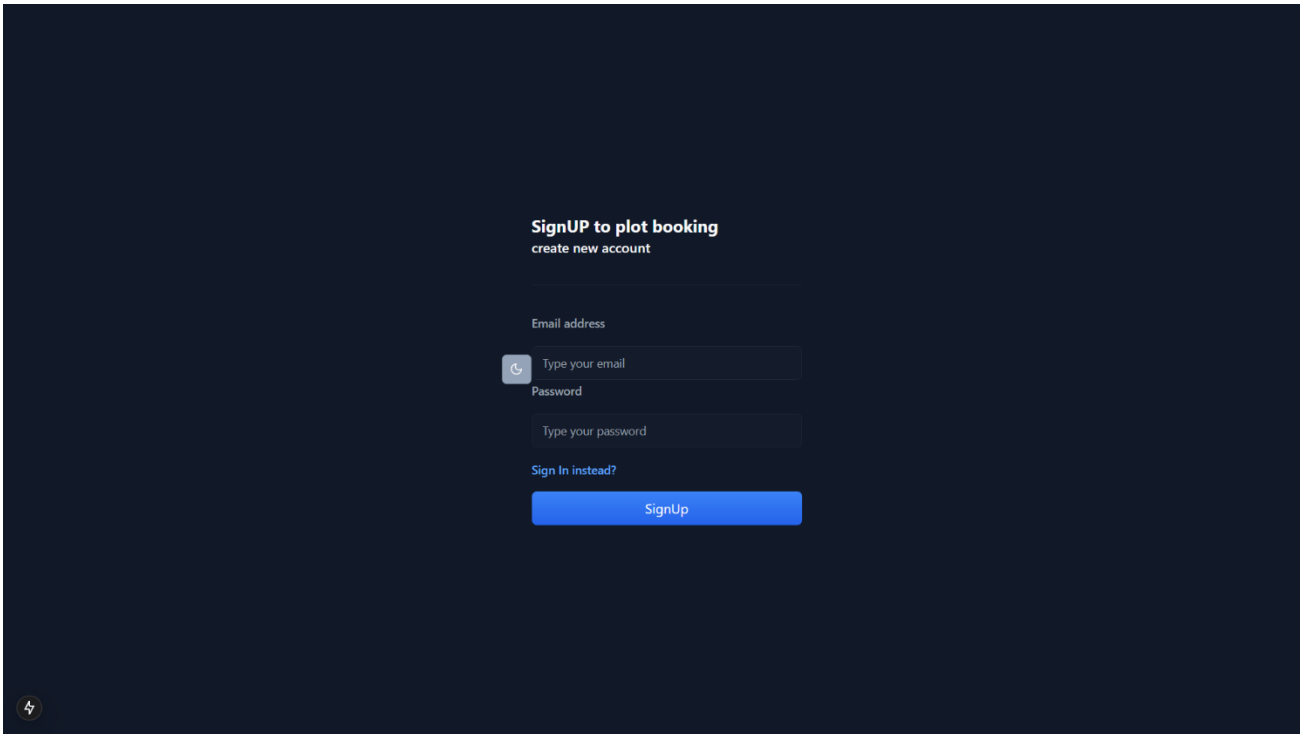


Figure 4.3-Sign Up Page of Plotify

2. Integration Testing
   Test the integration between frontend and backend, as well as payment processing

3. Functional Testing

   Objective: Test the functional requirements of the platform to ensure it performs as expected.

4.4 Screenshots of The Online Plot Booking System

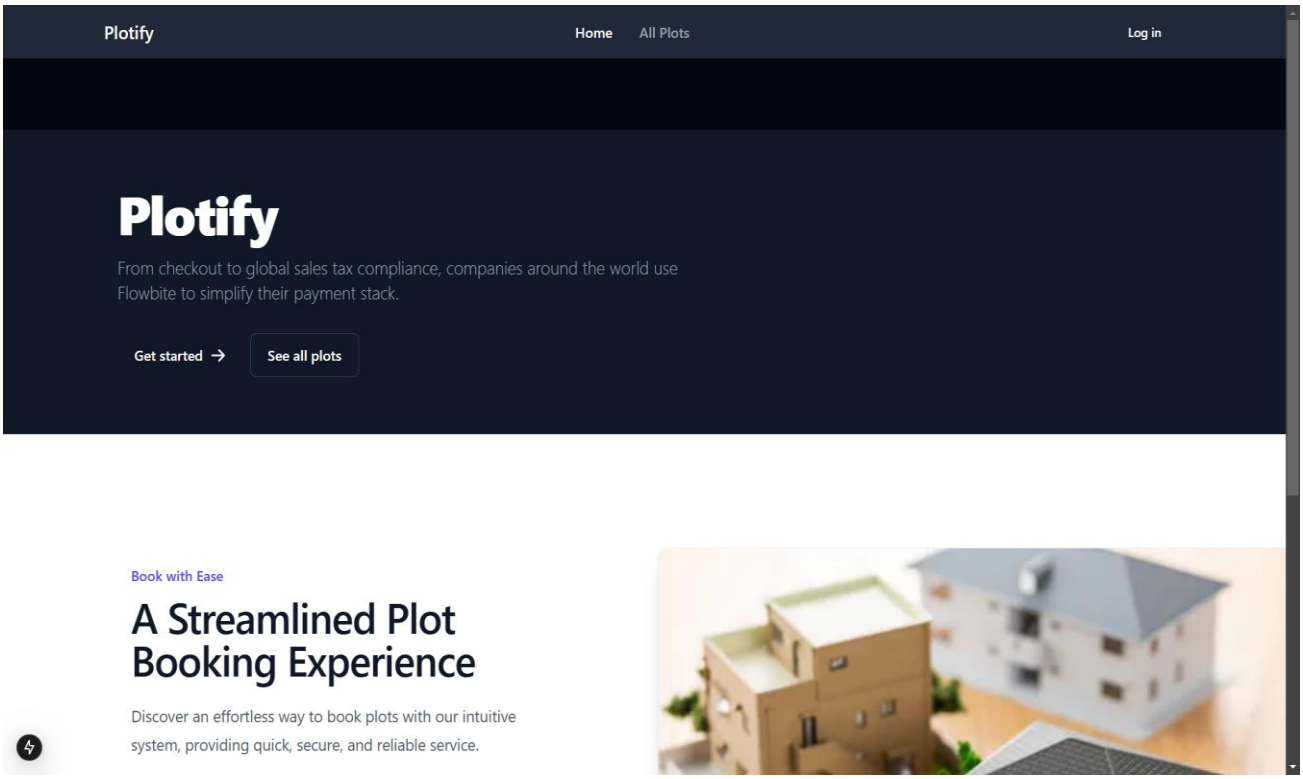


Figure 4.4-Home Page of Plotify

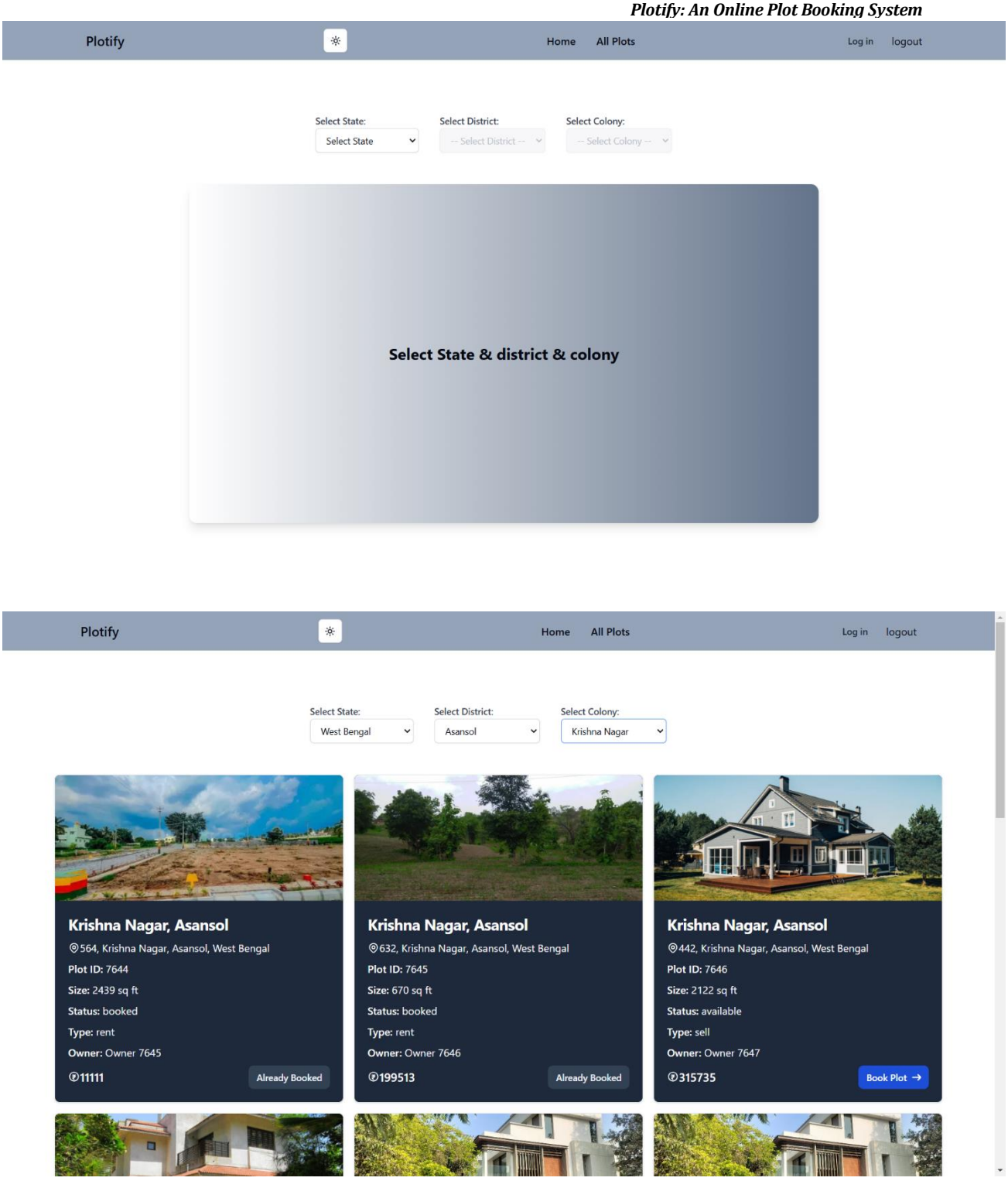


Figure 4.5-Plot Booking Page of Plotify

PLOTID	AREANO	STATE	DISTRICT	COLONY	DIMENSION	DIRECTION	STATUS	TYPE	PRICE	BOOKEDBY	OWNER	ADDRESS	ACTION
1	Area-16	Maharashtra	Pune	Shivaji Nagar	58 ft x 10 ft	East	available	rent	433388		Owner 2	575, Shivaji Nagar, Pune, Maharashtra	✕
2	Area-39	Maharashtra	Pune	Shivaji Nagar	41 ft x 28 ft	East	booked	rent	8450	ani@gmail.com	Owner 3	221, Shivaji Nagar, Pune, Maharashtra	✕
3	Area-4	Maharashtra	Pune	Shivaji Nagar	37 ft x 36 ft	South	available	sell	529003		Owner 4	770, Shivaji Nagar, Pune, Maharashtra	✕
4	Area-89	Maharashtra	Pune	Shivaji Nagar	64 ft x 30 ft	West	available	rent	316414		Owner 5	264, Shivaji Nagar, Pune, Maharashtra	✕
5	Area-84	Maharashtra	Pune	Shivaji Nagar	51 ft x 28 ft	South	available	rent	17360		Owner 6	833, Shivaji Nagar, Pune, Maharashtra	✕
6	Area-97	Maharashtra	Pune	Shivaji Nagar	61 ft x 36 ft	East	available	rent	6140		Owner 7	391, Shivaji Nagar, Pune, Maharashtra	✕
7	Area-19	Maharashtra	Pune	Shivaji Nagar	57 ft x 25 ft	South	booked	sell	304362		Owner 8	958, Shivaji Nagar, Pune, Maharashtra	✕
	Area-20	Maharashtra	Pune	Shivaji Nagar	66 ft x 17 ft	East	available	sell	18042		Owner 9	222, Shivaji Nagar, Pune, Maharashtra	✕

Figure 4.6-Admin Page of Plotify

Chapter 5. Conclusion

5.1 Conclusion

The development of the Online Plot Booking System continues to evolve with a focus on ensuring both functionality and scalability. As the platform progresses through various stages of development, it incorporates key user-centric features to enhance the overall experience. In the next phase of development, we will place a strong emphasis on refining the user interface (UI) and user experience (UX), ensuring that all design elements are intuitive and easy to navigate. This iterative process will involve gathering feedback from beta users to make real-time adjustments based on user needs and preferences, ultimately leading to a more personalized and effective platform.

Additionally, user authentication and account management features will be expanded, ensuring that both first-time users and returning customers have a seamless login experience. Secure login options, including multi-factor authentication (MFA), will be introduced to further safeguard user accounts and sensitive information, aligning with best practices for data security and privacy.

The system will also incorporate advanced search functionality, allowing users to filter available plots by various criteria such as size, location, price range, and other attributes. This flexibility will empower users to make informed decisions based on their specific needs, ultimately improving the overall satisfaction of users. Furthermore, the platform will be optimized for mobile devices, ensuring that the system is accessible on both desktop and mobile platforms, catering to users on the go.

To support growth and expansion, we will integrate additional features, such as a user feedback system that enables customers to rate and review plots and services. This will provide valuable insights into customer preferences and help administrators continuously improve the quality of listings and service offerings. The development of an admin dashboard for plot managers will allow for efficient management of listings, enabling easy updates to plot availability, pricing, and other essential details.

In parallel, a robust reporting and analytics tool will be implemented, enabling administrators to monitor usage patterns, identify trends, and make data-driven decisions to improve the platform's performance and user engagement. This data will also be useful in identifying potential areas for future enhancements and integrations.

Once the system has passed multiple rounds of rigorous functional, usability, and security testing, it will be ready for final deployment. The Online Plot Booking System will be optimized to handle increased traffic, ensuring that users experience fast load times and minimal downtime. A dedicated support and maintenance team will be established to monitor the system's performance, resolve issues promptly, and roll out regular updates to keep the platform secure and up to date.

In summary, the Online Plot Booking System is designed to provide an easy-to-use, secure, and efficient platform that meets the needs of both users and administrators. By focusing on scalability, security, and user experience, the platform will continue to evolve and improve, ensuring long-term success. The system offers a seamless booking process, real-time availability, and flexible search options, making it a valuable tool for users looking to secure plots with minimal hassle. Future enhancements, including customer feedback systems, mobile optimization, and advanced analytics, will continue to enhance the platform’s capabilities and ensure its relevance in a competitive market.

5.2 Limitations of the Work

1. Although the system is designed to handle multiple bookings, a sudden spike in user traffic may challenge performance without further optimization or load balancing.
2. A distributed database solution may be required as the system grows, especially for handling real-time data synchronization across multiple servers.
3. Implementing a payment gateway adds security risks and requires strict compliance with industry standards like PCI DSS (Payment Card Industry Data Security Standard).
4. Changes in payment provider APIs or payment failures can lead to disruptions and additional maintenance.
5. Although prototyping aids in understanding user interactions, certain technical details, such as response time and server load handling, cannot be fully simulated. These aspects may require further testing and optimization during actual development.
6. User data and transaction information are stored within the system, requiring stringent security protocols to prevent data breaches. Ensuring continuous compliance with data protection regulations (like GDPR) may be resource-intensive.
7. As an online service, the system relies on stable internet connections. This may limit access for users in areas with unreliable connectivity, impacting the system's accessibility and user experience.
8. The system does not provide personalized recommendations based on user behavior or preferences, which could enhance the user experience and help users discover plots of interest more effectively.
9. Implementing machine learning for recommendation would require further development and data analysis capabilities.

Bibliography

[1] https://www.magicbricks.com/

[2] https://github.com/pranavjain598/bookingsystem/

[3] https://keithweaverca.medium.com/using-socket-io-with-a-mern-stack-2a7049f94b85

[4] https://github.com/anitab-org/

[5] https://github.com/

[6] https://socket.io/

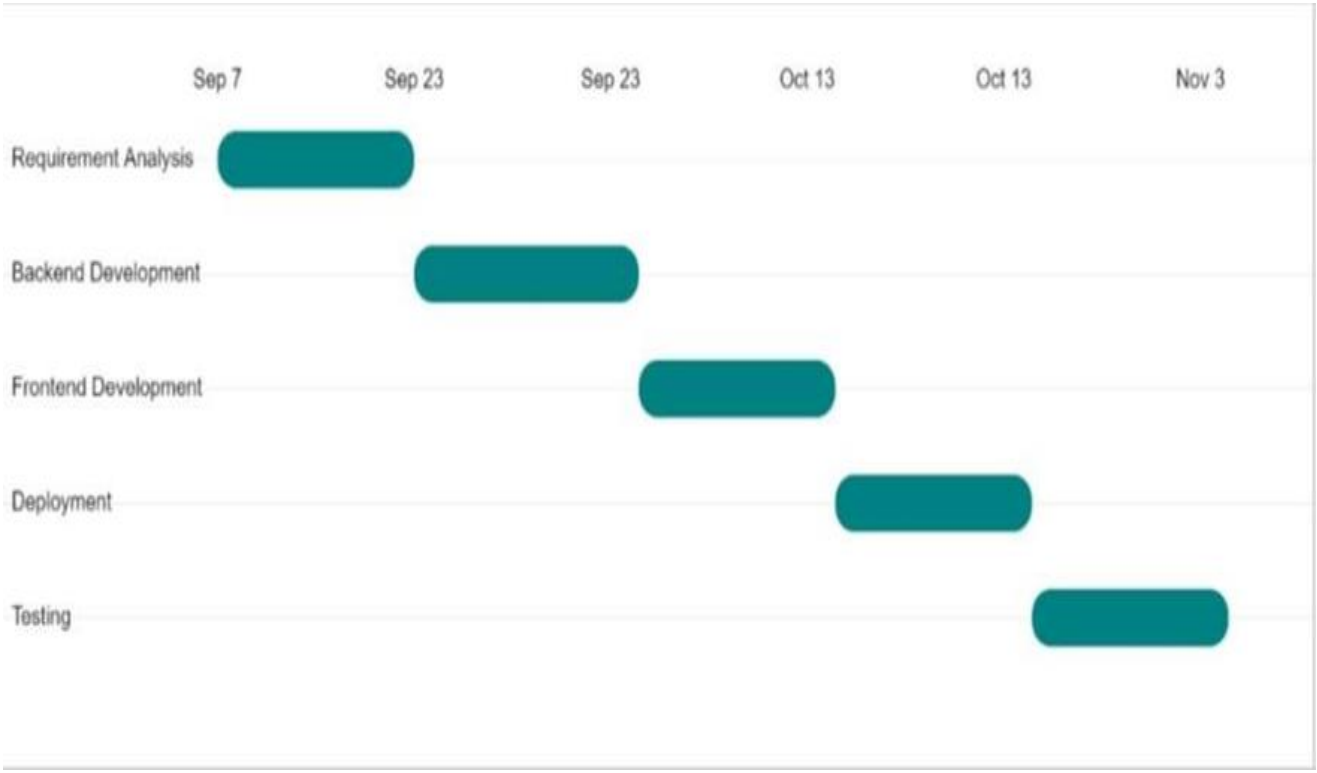
[7] https://www.geeksforgeeks.org/introduction-to-sockets-io-in-node-js/

[8] https://www.npmjs.com/package/socket.io

[9] https://www.mongodb.com/mern-stack

Appendix A

Project Plan



Guide Interaction Sheet

Date	Discussion	Action Plan
06/09/2024	Discussed project title selection.	Agreed upon "Plotify:Online Plot Booking System" as the project title.
13/09/2024	Discussed the approach for project development.	Selected Python for backend, Kivy for frontend, and Steganography.
20/09/2024	Presented synopsis and diagrams for review.	Reviewed and discussed necessary modifications for the synopsis.
1/10/2024	Presented PowerPoint on the project.	Conducted presentation on Plotify:Online Plot Booking System.
15/10/2024	Reviewed project implementation progress.	Discussed frontend and backend progress, identified necessary features.
25/10/2024	Presented and reviewed project report.	Reviewed the project report and outlined required revisions.
05/11/2024	Reviewed updated project implementation.	Reviewed frontend, backend, and discussed any remaining adjustments.
09/11/2024	Submission of Project synopsis, research paper, and report for review.	Submitted project documents for review and feedback.

Source Code

```
import Image from "next/image";
import React from "react";
// import homeImg from "../images/home";
import Link from "next/link";
import {
 CloudArrowUpIcon,
 LockClosedIcon,
 ServerIcon,
} from "@heroicons/react/20/solid";

const features = [
 {
  name: "Quick Booking Process",
  description:
   "Easily reserve your plot with a seamless booking process designed to save you time and effort.",
  icon: CloudArrowUpIcon,
 },
 {
  name: "Secure Transactions",
  description:
   "Enjoy peace of mind with SSL-secured transactions and data protection for all your booking details.",
  icon: LockClosedIcon,
 },
 {
  name: "Automated Land Records Backup",
  description:
   "Keep your records safe with regular backups, ensuring your data is always available when you need it.",
  icon: ServerIcon,
```

```
 },
];

const Home = () => {
 return (
  <div>
   <header className="mb-20">
    <nav className="dark:bg-white border-gray-200 px-4 lg:px-6 py-2.5 bg-gray-800 text-white">
     <div className="flex flex-wrap justify-between items-center mx-auto max-w-screen-xl">
      <a href="https://flowbite.com" className="flex items-center">
       <span className="self-center text-xl font-semibold whitespace-nowrap dark:text-white">
        Plotify
       </span>
      </a>
      <div className="flex items-center lg:order-2">
       <a
        href="/login"
        className="text-gray-800 dark:text-white hover:bg-gray-50 focus:ring-4 focus:ring-gray-300 font-medium rounded-lg text-sm px-4 lg:px-5 py-2 lg:py-2.5 mr-2 dark:hover:bg-gray-700 focus:outline-none dark:focus:ring-gray-800"
       >
        Log in
       </a>
       <button
        data-collapse-toggle="mobile-menu-2"
        type="button"
        className="inline-flex items-center p-2 ml-1 text-sm text-gray-500 rounded-lg lg:hidden hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-gray-200 dark:text-gray-400 dark:hover:bg-gray-700 dark:focus:ring-gray-600"
        aria-controls="mobile-menu-2"
        aria-expanded="false"
```

```
          >
            <span className="sr-only">Open main menu</span>
            <svg
              className="w-6 h-6"
              fill="currentColor"
              viewBox="0 0 20 20"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                fill-rule="evenodd"
                d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1
1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z"
                clip-rule="evenodd"
              ></path>
            </svg>
            <svg
              className="hidden w-6 h-6"
              fill="currentColor"
              viewBox="0 0 20 20"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                fill-rule="evenodd"
                d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414
1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-
1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z"
                clip-rule="evenodd"
              ></path>
            </svg>
          </button>
        </div>
        <div
          className="hidden justify-between items-center w-full lg:flex lg:w-auto lg:order-1"
          id="mobile-menu-2"
```

```
        >
          <ul className="flex flex-col mt-4 font-medium lg:flex-row lg:space-x-8 lg:mt-0">
           <li>
            <a
             href="home"
             className="block py-2 pr-4 pl-3 text-white rounded bg-primary-700 lg:bg-
transparent lg:text-primary-700 lg:p-0 dark:text-white"
             aria-current="page"
            >
             Home
            </a>
           </li>
           <li>
            <a
             href="/allplot"
             className="block py-2 pr-4 pl-3 border-b border-gray-100 hover:bg-gray-50
lg:hover:bg-transparent lg:border-0 lg:hover:text-primary-700 lg:p-0 dark:text-gray-400
lg:dark:hover:text-white dark:hover:bg-gray-700 dark:hover:text-white lg:dark:hover:bg-
transparent dark:border-gray-700"
            >
             All Plots
            </a>
           </li>
          </ul>
         </div>
        </div>
       </nav>
      </header>
      <section className="bg-white dark:bg-gray-900">
       <div className="grid max-w-screen-xl px-4 py-8 mx-auto lg:gap-8 xl:gap-0 lg:py-16
lg:grid-cols-12">
        <div className="mr-auto place-self-center lg:col-span-7">
         <h1 className="max-w-2xl mb-4 text-4xl font-extrabold tracking-tight leading-none
md:text-5xl xl:text-6xl dark:text-white">
```

```
          Plotify
        </h1>
        <p className="max-w-2xl mb-6 font-light text-gray-500 lg:mb-8 md:text-lg lg:text-xl
dark:text-gray-400">
          From checkout to global sales tax compliance, companies around the
          world use Flowbite to simplify their payment stack.
        </p>
        <a
          href="#"
          className="inline-flex items-center justify-center px-5 py-3 mr-3 text-base font-
medium text-center text-white rounded-lg bg-primary-700 hover:bg-primary-800
focus:ring-4 focus:ring-primary-300 dark:focus:ring-primary-900"
        >
          Get started
          <svg
            className="w-5 h-5 ml-2 -mr-1"
            fill="currentColor"
            viewBox="0 0 20 20"
            xmlns="http://www.w3.org/2000/svg"
          >
            <path
              fill-rule="evenodd"
              d="M10.293 3.293a1 1 0 011.414 0l6 6a1 1 0 010 1.414l-6 6a1 1 0 01-1.414-
1.414L14.586 11H3a1 1 0 110-2h11.586l-4.293-4.293a1 1 0 010-1.414z"
              clip-rule="evenodd"
            ></path>
          </svg>
        </a>
        <Link
          href="/allplot"
          className="inline-flex items-center justify-center px-5 py-3 text-base font-medium
text-center text-gray-900 border border-gray-300 rounded-lg hover:bg-gray-100
focus:ring-4 focus:ring-gray-100 dark:text-white dark:border-gray-700 dark:hover:bg-
gray-700 dark:focus:ring-gray-800"
```

```
        >
          See all plots
        </Link>
       </div>
       <div className="hidden lg:mt-0 lg:col-span-5 lg:flex">
        {/* <Image src={homeImg} alt="mockup" /> */}
       </div>
      </div>
    </section>

    {/* second start  */}
    <div className="overflow-hidden bg-white py-24 sm:py-32">
     <div className="mx-auto max-w-7xl px-6 lg:px-8">
      <div className="mx-auto grid max-w-2xl grid-cols-1 gap-x-8 gap-y-16 sm:gap-y-20
lg:mx-0 lg:max-w-none lg:grid-cols-2">
       <div className="lg:pr-8 lg:pt-4">
        <div className="lg:max-w-lg">
         <h2 className="text-base/7 font-semibold text-indigo-600">
          Book with Ease
         </h2>
         <p className="mt-2 text-4xl font-semibold tracking-tight text-gray-900 sm:text-
5xl">
          A Streamlined Plot Booking Experience
         </p>
         <p className="mt-6 text-lg/8 text-gray-600">
          Discover an effortless way to book plots with our intuitive
          system, providing quick, secure, and reliable service.
         </p>
         <dl className="mt-10 max-w-xl space-y-8 text-base/7 text-gray-600 lg:max-w-
none">
          {features.map((feature) => (
           <div key={feature.name} className="relative pl-9">
            <dt className="inline font-semibold text-gray-900">
             <feature.icon
```

```
                aria-hidden="true"
                className="absolute left-1 top-1 h-5 w-5 text-indigo-600"
               />
               {feature.name}
              </dt>{" "}
              <dd className="inline">{feature.description}</dd>
             </div>
            ))}
           </dl>
          </div>
         </div>
         <img
          alt="Plot booking system screenshot"
          src="https://imgs.search.brave.com/VZ0qY_HZn6S-
WikV0F737mRBeleQS3qTQ4jVceM-
3qw/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMT
E4/MDkzMTM3Mi9waG90/by9ob3VzZS1tb2Rl/bC1vbi13b29kZW4t/dGFibGUuanBnP3M9
/NjEyeDYxMiZ3PTAm/az0yMCZjPTR1NmRB/WmU0M0M1TElJc2VS/VFRVWDZGbTNaaGp
M/cGphb01Ib3h5U2xS/a009"
          width={2432}
          height={1442}
          className="w-[48rem] max-w-none rounded-xl shadow-xl ring-1 ring-gray-
400/10 sm:w-[57rem] md:-ml-4 lg:-ml-0"
         />
        </div>
       </div>
      </div>
     </div>
    );
};

export default Home;
```

```
import { createMiddlewareClient } from "@supabase/auth-helpers-nextjs";
import { NextRequest, NextResponse } from "next/server";
```

```
export async function middleware(req: NextRequest) {
  const res = NextResponse.next();
  const supabase = createMiddlewareClient({ req, res });
  const {
    data: { session },
  } = await supabase.auth.getSession();
  console.log(session);
  if(!session){
    return NextResponse.rewrite(new URL('/unauthenticated', req.url))
  }


  return res;
}

export const config = {
  matcher: [
    "/authenticated", "/allplot" ,"/bookPlot"
  ],
};

import React from 'react';

interface ButtonProps {
  status: string;
}

const Button: React.FC<ButtonProps> = ({ status }) => {
  return ( status==="paid"?
    <div className="relative inline-flex items-center justify-center overflow-hidden
rounded-lg bg-gradient-to-b from-[#3ffd68] to-[#18f888] transition-all duration-200 ease-
linear hover:scale-105">
      <button
        type="button"
        className="relative inline-flex items-center justify-center h-10 w-20 overflow-hidden
rounded-md bg-gradient-to-b from-[#71ff94] to-[#71ffbd] transition-all duration-300
ease-in-out hover:scale-95"
      >
        <span className="absolute top-0 right-0 h-4 w-4 rounded-tr-md rounded-bl-md bg-
gradient-to-b from-[#71ff89] to-[#71ffa7] shadow-md transition-all duration-500 ease-in-
out transform-gpu hover:-translate-y-4 hover:-translate-x-4"></span>

        <div className="absolute inset-0 overflow-hidden pointer-events-none z-0">
          {Array.from({ length: 10 }).map((_, i) => (
            <i
              key={i}
              className={`absolute bottom-0 h-[2px] w-[2px] bg-blue-600 rounded-full
animate-floating-points delay-[${i * 0.2}s]`}
              style={{ left: `${Math.random() * 100}%`, animationDuration: `${1.5 +
Math.random()}s` }}
```

```
        ></i>
       ))}
      </div>

      <span className="relative z-10 flex items-center gap-1 text-white font-medium text-
lg">
       <svg
        className="w-5 h-5 text-white transition-colors duration-200 ease-in-out"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
       >
        <polyline
         points="13.18 1.37 13.18 9.64 21.45 9.64 10.82 22.63 10.82 14.36 2.55 14.36 13.18
1.37"
         className="stroke-[2.5] stroke-current transition-all duration-700 ease-in-out"
        />
       </svg>
       {status}
      </span>
    </button>
   </div>:
   <div className="relative inline-flex items-center justify-center overflow-hidden
rounded-lg bg-gradient-to-b from-[#f34c4c] to-[#fa9f49] transition-all duration-200 ease-
linear hover:scale-105">
   <button
    type="button"
    className="relative inline-flex items-center justify-center h-10 w-20 overflow-hidden
rounded-md bg-gradient-to-b from-[#ff8466] to-[#f86945] transition-all duration-300
ease-in-out hover:scale-95"
   >
    <span className="absolute top-0 right-0 h-4 w-4 rounded-tr-md rounded-bl-md bg-
gradient-to-b from-[#fa5454] to-[#f58e19] shadow-md transition-all duration-500 ease-
in-out transform-gpu hover:-translate-y-4 hover:-translate-x-4"></span>

    <div className="absolute inset-0 overflow-hidden pointer-events-none z-0">
     {Array.from({ length: 10 }).map((_, i) => (
      <i
       key={i}
       className={`absolute bottom-0 h-[2px] w-[2px] bg-blue-600 rounded-full animate-
floating-points delay-[${i * 0.2}s]`}
       style={{ left: `${Math.random() * 100}%`, animationDuration: `${1.5 +
Math.random()}s` }}
      ></i>
     ))}
    </div>

    <span className="relative z-10 flex items-center gap-1 text-white font-medium text-
base">
     <svg
```

```
        className="w-5 h-5 hover:fill-white text-white transition-colors duration-200 ease-
in-out"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
       >
        <polyline
         points="13.18 1.37 13.18 9.64 21.45 9.64 10.82 22.63 10.82 14.36 2.55 14.36 13.18
1.37"
         className="stroke-[2.5] stroke-current transition-all duration-700 ease-in-out"
        />
       </svg>
       {status}
      </span>
    </button>
  </div>
  );
};

export default Button;

"use client"
import React from "react";
import { createClient } from "@/utils/supabase/client";

const Table = ({ data }) => {
  const supabase = createClient();

  const handleRemoveBooking = async (plotId) => {
    try {
      const { data, error } = await supabase
        .from("allplot")
        .update({
          bookedBy: "",    // Clear the bookedBy field
          status: "available", // Change the status to "booked"
        })
        .eq("plotId", plotId);

      if (error) {
        console.error("Error updating booking:", error);
      } else {
        console.log("Booking removed for plotId:", plotId);
        // Reload the page to reflect the update
        window.location.reload();
      }
    } catch (err) {
      console.error("Unexpected error:", err);
    }
  };
```

```
 return (
  <div className="overflow-x-auto shadow-md sm:rounded-lg">
   <table className="w-full text-sm text-left rtl:text-right text-gray-500 dark:text-gray-
400">
    <thead className="text-xs text-gray-700 uppercase bg-gray-50 dark:bg-gray-700
dark:text-gray-400">
     <tr>
      <th scope="col" className="px-6 py-3">plotId</th>
      <th scope="col" className="px-6 py-3">areaNo</th>
      <th scope="col" className="px-6 py-3">state</th>
      <th scope="col" className="px-6 py-3">district</th>
      <th scope="col" className="px-6 py-3">colony</th>
      <th scope="col" className="px-6 py-3">dimension</th>
      <th scope="col" className="px-6 py-3">direction</th>
      <th scope="col" className="px-6 py-3">status</th>
      <th scope="col" className="px-6 py-3">type</th>
      <th scope="col" className="px-6 py-3">price</th>
      <th scope="col" className="px-6 py-3">bookedBy</th>
      <th scope="col" className="px-6 py-3">owner</th>
      <th scope="col" className="px-6 py-3">address</th>
      <th scope="col" className="px-6 py-3">Action</th>
     </tr>
    </thead>

    <tbody>
     {data.map((el) => (
      <tr key={el.plotId}>
       <td className="px-6 py-4 font-medium text-gray-900 whitespace-nowrap
dark:text-white">{el.plotId}</td>
       <td className="px-6 py-4">{el.areaNo}</td>
       <td className="px-6 py-4">{el.state}</td>
       <td className="px-6 py-4">{el.district}</td>
       <td className="px-6 py-4">{el.colony}</td>
       <td className="px-6 py-4">{el.dimension}</td>
       <td className="px-6 py-4">{el.direction}</td>
       <td className="px-6 py-4">{el.status}</td>
       <td className="px-6 py-4">{el.type}</td>
       <td className="px-6 py-4">{el.price}</td>
       <td className="px-6 py-4">{el.bookedBy}</td>
       <td className="px-6 py-4">{el.owner}</td>
       <td className="px-6 py-4">{el.address}</td>
       <td className="px-6 py-4">
        <button
         className="bg-slate-200 p-2 hover:bg-slate-500"
         onClick={() => handleRemoveBooking(el.plotId)}
        >
         ❌
        </button>
       </td>
      </tr>
     ))}
```

```
      </tbody>
     </table>
   </div>
  );
};

export default Table;
import React from "react";

const CurrencyRupee = ({ size = 24 }) => (
 <svg
  xmlns="http://www.w3.org/2000/svg"
  width={size}
  height={size}
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
  strokeWidth={2}
 >
  <path
   strokeLinecap="round"
   strokeLinejoin="round"
   d="M9 8h6m-5 0a3 3 0 110 6H9l3 3m-3-6h6m6 0a9 9 0 11-18 0 9 9 0 0118 0z"
  />
 </svg>
);

const MapPin = ({ size = 24 }) => (
 <svg
  xmlns="http://www.w3.org/2000/svg"
  width={size}
  height={size}
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
  strokeWidth={2}
 >
  <path
   strokeLinecap="round"
   strokeLinejoin="round"
   d="M17.657 16.657L13.414 20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8 8 0
1111.314 0z"
  />
  <path
   strokeLinecap="round"
   strokeLinejoin="round"
   d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"
  />
 </svg>
);
```

```
const PropertyCard = ({
  imageUrl,
  plotId,
  state,
  district,
  colony,
  size,
  status,
  type,
  price,
  bookedBy,
  owner,
  address,
}) => {
  return (
    <div className="bg-white border border-gray-200 rounded-lg shadow dark:bg-gray-
800 dark:border-gray-700 overflow-hidden">
      <img

src="https://imgs.search.brave.com/ccAPasDnjJWl_RCs2eCiQacz6omap0M1VXPp_sN5qJU
/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9pbWcu/c3RhdGljbWIuY29t/L21icGhvdG8vcHJv/cGV
ydHkvY3JvcHBl/ZF9pbWFnZXMvMjAy/NC9TZXAvMTEvUGhv/dG9faDMwMF93NDUw/Lzc
c0ODQ3OTI1XzJf/UHJvcGVydHlJbWFn/ZTExMy04OTk3NTAy/Mjc1MTQ0Ml8zMDBf/NDU
wLmpwZw"
        alt="Property"
        className="w-full h-48 object-cover"
      />
      <div className="p-5">
        <h5 className="mb-2 text-2xl font-bold tracking-tight text-gray-900 dark:text-
white">
          {colony}, {district}
        </h5>
        <div className="flex items-center mb-2">
          <MapPin size={18} className="mr-2" />
          <span>{address}</span>
        </div>
        <div className="mb-2">
          <span className="font-medium">Plot ID:</span> {plotId}
        </div>
        <div className="mb-2">
          <span className="font-medium">Size:</span> {size}
        </div>
        <div className="mb-2">
          <span className="font-medium">Status:</span> {status}
        </div>
        <div className="mb-2">
          <span className="font-medium">Type:</span> {type}
        </div>
        <div className="mb-2">
          <span className="font-medium">Owner:</span> {owner}
        </div>
```

```
      {bookedBy && (
       <div className="mb-2">
        <span className="font-medium">Booked By:</span> {bookedBy}
       </div>
      )}
      <div className="flex justify-between items-center">
       <div className="flex items-center">
        <CurrencyRupee size={18} className="mr-2" />
        <span className="text-lg font-medium">{price}</span>
       </div>
       <a
        href="#"
        className="inline-flex items-center px-3 py-2 text-sm font-medium text-center
text-white bg-blue-700 rounded-lg hover:bg-blue-800 focus:ring-4 focus:outline-none
focus:ring-blue-300 dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"
       >
        Book Plot
        <svg
         className="rtl:rotate-180 w-3.5 h-3.5 ms-2"
         aria-hidden="true"
         xmlns="http://www.w3.org/2000/svg"
         fill="none"
         viewBox="0 0 14 10"
        >
         <path
          stroke="currentColor"
          strokeLinecap="round"
          strokeLinejoin="round"
          strokeWidth={2}
          d="M1 5h12m0 0L9 1m4 4L9 9"
         />
        </svg>
       </a>
      </div>
     </div>
    </div>
  );
};
const Example = () => {
 const data = [
  {
   plotId: 1,
   state: "Maharashtra",
   district: "Pune",
   colony: "Shivaji Nagar",
   size: "636 sq ft",
   status: "available",
   type: "sell",
   price: 11865,
   bookedBy: null,
   owner: "Owner 2",
```

```
  address: "240, Shivaji Nagar, Pune, Maharashtra",
 },
 {
  plotId: 2,
  state: "Maharashtra",
  district: "Pune",
  colony: "Gokul Colony",
  size: "1343 sq ft",
  status: "booked",
  type: "rent",
  price: 468043,
  bookedBy: null,
  owner: "Owner 3",
  address: "288, Gokul Colony, Pune, Maharashtra",
 },
 {
  plotId: 3,
  state: "Maharashtra",
  district: "Pune",
  colony: "Raja Park",
  size: "502 sq ft",
  status: "booked",
  type: "rent",
  price: 23677,
  bookedBy: null,
  owner: "Owner 4",
  address: "577, Raja Park, Pune, Maharashtra",
 },
 {
  plotId: 4,
  state: "Maharashtra",
  district: "Pune",
  colony: "Krishna Nagar",
  size: "2070 sq ft",
  status: "booked",
  type: "rent",
  price: 276793,
  bookedBy: "dxtpa@example.com",
  owner: "Owner 5",
  address: "484, Krishna Nagar, Pune, Maharashtra",
 },
 {
  plotId: 5,
  state: "Maharashtra",
  district: "Pune",
  colony: "Green Valley",
  size: "1371 sq ft",
  status: "booked",
  type: "sell",
  price: 427363,
  bookedBy: "pohda@example.com",
```

```
  owner: "Owner 6",
  address: "685, Green Valley, Pune, Maharashtra",
 },
 {
  plotId: 6,
  state: "Maharashtra",
  district: "Pune",
  colony: "Sunrise Enclave",
  size: "2270 sq ft",
  status: "booked",
  type: "sell",
  price: 16273,
  bookedBy: null,
  owner: "Owner 7",
  address: "850, Sunrise Enclave, Pune, Maharashtra",
 },
 {
  plotId: 7,
  state: "Maharashtra",
  district: "Pune",
  colony: "Pearl Residency",
  size: "1689 sq ft",
  status: "available",
  type: "rent",
  price: 104516,
  bookedBy: "vm2sk@example.com",
  owner: "Owner 8",
  address: "610, Pearl Residency, Pune, Maharashtra",
 },
 {
  plotId: 8,
  state: "Maharashtra",
  district: "Pune",
  colony: "Elite Garden",
  size: "1195 sq ft",
  status: "available",
  type: "rent",
  price: 575678,
  bookedBy: null,
  owner: "Owner 9",
  address: "67, Elite Garden, Pune, Maharashtra",
 },
 {
  plotId: 9,
  state: "Maharashtra",
  district: "Pune",
  colony: "Silver Oak",
  size: "730 sq ft",
  status: "booked",
  type: "sell",
  price: 369751,
```

```
  bookedBy: "tr5gi@example.com",
  owner: "Owner 10",
  address: "794, Silver Oak, Pune, Maharashtra",
 },
 {
  plotId: 10,
  state: "Maharashtra",
  district: "Pune",
  colony: "Maple Heights",
  size: "734 sq ft",
  status: "booked",
  type: "sell",
  price: 266087,
  bookedBy: null,
  owner: "Owner 11",
  address: "117, Maple Heights, Pune, Maharashtra",
 },
 {
  plotId: 11,
  state: "Maharashtra",
  district: "Mumbai",
  colony: "Shivaji Nagar",
  size: "2241 sq ft",
  status: "booked",
  type: "rent",
  price: 6173,
  bookedBy: "azxh9@example.com",
  owner: "Owner 12",
  address: "110, Shivaji Nagar, Mumbai, Maharashtra",
 },
 {
  plotId: 12,
  state: "Maharashtra",
  district: "Mumbai",
  colony: "Gokul Colony",
  size: "1997 sq ft",
  status: "available",
  type: "sell",
  price: 23813,
  bookedBy: null,
  owner: "Owner 13",
  address: "967, Gokul Colony, Mumbai, Maharashtra",
 },
 {
  plotId: 13,
  state: "Maharashtra",
  district: "Mumbai",
  colony: "Raja Park",
  size: "1019 sq ft",
  status: "available",
  type: "sell",
```

```
    price: 17189,
    bookedBy: null,
    owner: "Owner 14",
    address: "276, Raja Park, Mumbai, Maharashtra",
   },
  ];

  return (
   <div>
    <header className="mb-20">
     <nav className="bg-white border-gray-200 px-4 lg:px-6 py-2.5 dark:bg-gray-800">
      <div className="flex flex-wrap justify-between items-center mx-auto max-w-screen-
xl">
       <a href="" className="flex items-center">
        <span className="self-center text-xl font-semibold whitespace-nowrap dark:text-
white">
         Plotify
        </span>
       </a>
       <div className="flex items-center lg:order-2">
        <a
         href="/login"
         className="text-gray-800 dark:text-white hover:bg-gray-50 focus:ring-4
focus:ring-gray-300 font-medium rounded-lg text-sm px-4 lg:px-5 py-2 lg:py-2.5 mr-2
dark:hover:bg-gray-700 focus:outline-none dark:focus:ring-gray-800"
        >
         Log in
        </a>
        <button
         data-collapse-toggle="mobile-menu-2"
         type="button"
         className="inline-flex items-center p-2 ml-1 text-sm text-gray-500 rounded-lg
lg:hidden hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-gray-200
dark:text-gray-400 dark:hover:bg-gray-700 dark:focus:ring-gray-600"
         aria-controls="mobile-menu-2"
         aria-expanded="false"
        >
         <span className="sr-only">Open main menu</span>
         <svg
          className="w-6 h-6"
          fill="currentColor"
          viewBox="0 0 20 20"
          xmlns="http://www.w3.org/2000/svg"
         >
          <path
           fill-rule="evenodd"
           d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1
1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z"
           clip-rule="evenodd"
          ></path>
         </svg>
```

```
            <svg
              className="hidden w-6 h-6"
              fill="currentColor"
              viewBox="0 0 20 20"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                fill-rule="evenodd"
                d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414
1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-
1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z"
                clip-rule="evenodd"
              ></path>
            </svg>
          </button>
        </div>
        <div
          className="hidden justify-between items-center w-full lg:flex lg:w-auto lg:order-
1"
          id="mobile-menu-2"
        >
          <ul className="flex flex-col mt-4 font-medium lg:flex-row lg:space-x-8 lg:mt-0">
            <li>
              <a
                href="/home"
                className="block py-2 pr-4 pl-3 text-white rounded bg-primary-700 lg:bg-
transparent lg:text-primary-700 lg:p-0 dark:text-white"
                aria-current="page"
              >
                Home
              </a>
            </li>
            <li>
              <a
                href="allplot"
                className="block py-2 pr-4 pl-3 text-gray-700 border-b border-gray-100
hover:bg-gray-50 lg:hover:bg-transparent lg:border-0 lg:hover:text-primary-700 lg:p-0
dark:text-gray-400 lg:dark:hover:text-white dark:hover:bg-gray-700 dark:hover:text-
white lg:dark:hover:bg-transparent dark:border-gray-700"
              >
                All Plots
              </a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>
  <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {data.map((item) => (
      <PropertyCard
```

```
          key={item.plotId}
          imageUrl={`/api/placeholder/400/240?id=${item.plotId}`}
          plotId={item.plotId}
          state={item.state}
          district={item.district}
          colony={item.colony}
          size={item.size}
          status={item.status}
          type={item.type}
          price={item.price}
          bookedBy={item.bookedBy}
          owner={item.owner}
          address={item.address}
        />
      ))}
    </div>
  </div>
 );
};

export default Example;
```

Appendix B- Plotify User Manual

1. Introduction:

Plotify is a cutting-edge application designed to encode and decode secret messages within digital media files. Whether you're looking to send encrypted messages or decode secure content, Plotify provides an intuitive and reliable platform for your needs. Perfect for personal use, corporate security, or creative communication.

1.1. System Requirements:

- Operating System: Windows 10+, macOS 11+, or Linux (Ubuntu 20.04+ recommended)
- RAM: Minimum 4GB
- Storage: 500MB of free space
- Additional: Python 3.8+ installed (if using CLI version)

2. Installation

- Download Plotify:
- Visit the official Plotify website or download from an app store.
- Install Application:
- Run the installer and follow the on-screen instructions.
- Launch Plotify:

- Open the app and set up your profile for the first use.
- For advanced users, Plotify offers a command-line version. Use pip install plotify to get started.

3. Features Overview

1. Message Encoding: Hide messages within images, audio, or video files securely.
2. Message Decoding: Extract secret messages from encoded files.
3. Multiple Encoding Methods: Choose from various algorithms (e.g., LSB, DCT).
4. File Type Support: Compatible with .png, .jpg, .wav, and more.
5. User-Friendly UI: Easy-to-use graphical interface.

4. User Interface Walkthrough

4.1. Main Dashboard:

- Encode Button: Select to start encoding a message.
- Decode Button: Begin decoding a message from a file.
- Settings Icon: Customize encryption methods, themes, and more.

4.2. Encoding Screen:

- File Input Field: Upload the media file you wish to encode.
- Message Input Box: Type the message to encode.
- Encrypt Options: Add a password (optional).

4.3. Decoding Screen:

- File Input Field: Upload the encoded media file.
- Password Field: Enter the decryption password (if set).
- Decode Button: Extract the hidden message.

5. How to Encode a Message

5.1. Open the Encoding Screen from the main dashboard.

5.2. Click Upload File and select the media file.

5.3. Type your secret message in the Message Box.

5.4. (Optional) Set a password for encryption.

5.5. Click Encode.

5.6. Save the encoded file.

6. How to Decode a Message

6.1. Navigate to the Decoding Screen.

6.2. Upload the media file with the encoded message.

6.3. If a password was set, enter it in the Password Field.

6.4. Click Decode.

6.5. View the extracted message.

7. Best Practices and Tips

7.1. Use High-Quality Media Files: Higher resolution files ensure better encoding without noticeable changes.

7.2. Choose Strong Passwords: To maximize security, use a mix of letters, numbers, and symbols.

7.3. Test Before Sharing: Decode your file to ensure accuracy before sending it.

7.4. Keep Files Confidential: Share files through secure platforms.

8. Frequently Asked Questions (FAQ)

8.1. What file types are supported?

- Plotify supports common image formats (e.g., .png, .jpg), audio (.wav), and more.

8.2. Can I decode without the password?

- No, you must know the password if one was set.

8.3. Is Plotify free?

- **Plotify offers both free and premium versions.**

9. Troubleshooting

9.1. Error: "Unsupported File Type":

- Ensure you're using a compatible file format.

9.2. Message Decoding Fails:

- Verify the file wasn't modified after encoding.
- Check for correct password usage.

9.3. Application Crashes:

- Update to the latest version or reinstall the app.

10. Contact and Support

10.1. Email: support@plotify.com

10.2. Phone: +1-800-PLOTIFY

10.3. Community Forum: Plotify Support Forum

Appendix C- Research Paper

Reseach paper