PROJECT REPORT

On

RENTAL WEBSITE

Submitted in partial fulfilment of the requirement for the
Course BEE (22CS026) of

COMPUTER SCIENCE AND ENGINEERING
B.E. Batch-2022

.
Ill

Jan -2025

Under the Guidance of
Ms. Shalini
Mentor

Submitted By

Mohit Kumar
Roll. No. 2210991931
Moksh Goyal
Roll. No.2210991933
Muddit Pahwa
Roll. No.2210991942
Navam Sharma
Roll. No. 2210991961
Nikhar Raj
Roll. No. 2210991976

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHITKARA UNIVERSITY
PUNJAB

CHITKARA UNIVERSITY

CERTIFICATE

This is to be certified that the project entitled "Rental Website" has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2024-May-2024 is a bonafide piece of project work carried out by Mohit Kumar (2210991931), Moksh Goyal (2210991933), Muddit Pahwa (2210991942), Navam Sharma (221099161), Nikhar Raj (2210991976) towards the partial fulfillment for the award of the course Integrated Project (CS 203) under the guidance of Ms, Shalini and supervision.

Sign. of Project Guide

: Ms. Shalini

(Mentor)

CHITKARA
UNIVERSITY

CANDIDATE'S DECLARATION

We, **Mohit Kumar (2210991931), Moksh Goyal (2210991933), Muddit Pahwa (2210991942), Navam Sharma (221099161), Nikhar Raj (2210991976),** B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled **"RENTAL WEBSITE"** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Sign. of Student 1	**Sign. of Student 2**	**Sign. of Student 3**
Mohit Kumar	Moksh Goyal	Muddit Pahwa
2210991931	2210991933	2210991942
Sign. of Student 4	**Sign. of Student 5**	
Navam Sharma	Nikhar Raj	
2210991961	2210991976	

Place:
Date:

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as the part of the curriculum.

We are thankful to Ms. Shalini for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to **Ms. Shalini** who provided his valuable suggestions and precious time in accomplishing our Integrated project report.

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

Mohit Kumar

2210991931

Moksh Goyal

2210991933

Muddit Pahwa

2210991942

Navam Sharma

2210991961

Nikhar Raj

2210991976

ABSTRACT

This project focuses on developing a robust real estate platform using the MERN stack (MongoDB, Express.js, React.js, Node.js) to streamline property transactions for buyers, sellers, and agents. It incorporates key features such as secure user authentication via JWT, profile management for personal details and property preferences, and property listings with comprehensive information like type, price, location, and photos. The platform allows users to search for properties using advanced filters and sorting options based on criteria like location, price, and amenities, ensuring a seamless and efficient search experience. Detailed property pages provide in-depth descriptions and images to help users make informed decisions.

The platform is designed with a responsive, user-friendly interface using React.js to ensure an intuitive experience across devices. MongoDB powers the scalable backend, enabling efficient data management and supporting the growing user base and listings. Future enhancements will include features such as real-time chat, saved searches, alerts, and integrated payment options, further improving the overall user experience and making property transactions easier for all stakeholders involved.

1. INTRODUCTION TO THE PROJECT

1.1 BACKGROUND

The real estate industry is undergoing a significant digital transformation as buyers, sellers, and agents increasingly rely on online platforms to facilitate property transactions. Traditional methods of real estate dealing—such as classified ads, physical visits, and manual documentation—are becoming inefficient in a fast-paced, technology-driven world. Today's users demand convenience, transparency, and efficiency when searching for, listing, or managing properties. However, existing digital platforms often fall short due to outdated interfaces, limited functionality, and poor user experiences.

As the demand for streamlined and user-centric online property management grows, there is a need for a more modern solution that integrates various functionalities into a single, cohesive platform. Leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js) presents an opportunity to develop a powerful, scalable, and interactive real estate web application that caters to the evolving needs of all stakeholders—buyers, sellers, and agents—while ensuring high performance, flexibility, and future adaptability.

1.2 PROBLEM STATEMENT

Current real estate websites lack comprehensive features, user-friendly interfaces, efficient data management, and robust security. Many platforms offer only basic property listings and searches without advanced filtering or profile management. They often have outdated designs, poor responsiveness, and limited communication tools, making the user experience frustrating and inefficient. Additionally, scalability issues arise with growing data, and weak security measures expose users to risks.

There is a need for a modern real estate platform that integrates secure user authentication, detailed property management, advanced search and filtering, and real-time interactions within a scalable, user-friendly environment. This project aims to address these gaps using the MERN stack to create an efficient and adaptable solution.

2. SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATION

2.1 METHODS

The MERN Stack Framework will be the core technology stack for building the application. This includes MongoDB for a flexible and scalable NoSQL database, Express.js to manage server-side logic and API development, React.js to create a dynamic and responsive user interface, and Node.js for running JavaScript on the server side to ensure high performance and scalability.

To maintain efficient collaboration and control over the development process, version control with Git will be employed. GitHub or GitLab will serve as the repository management system, supporting branching strategies like "feature branching" or "Gitflow" to streamline development, manage code changes, and avoid conflicts. These platforms will also be used for code reviews, issue tracking, and continuous integration to ensure smooth and organized teamwork.

Security best practices will be a core focus throughout the development process. The project will use JWT for secure user authentication, HTTPS for encrypted communication, and robust input validation to prevent vulnerabilities like SQL injection, XSS, and CSRF. Regular security audits and penetration testing will be conducted to identify and mitigate potential risks, ensuring a secure and reliable platform for all users.

2.2 PROGRAMMING/WORKING ENVIRONMENT

The development of the real estate platform will be conducted in a structured environment to ensure efficiency and consistency. Development will take place on cross-platform operating systems like Windows, Linux, or macOS, providing flexibility and compatibility for all team members.

Visual Studio Code (VS Code) will be the primary code editor due to its extensive support for JavaScript and Node.js, along with its productivity-enhancing extensions. Node.js will serve as the runtime environment, with NPM managing project dependencies such as Express.js and React.js. For database management, MongoDB will be used, with tools like MongoDB Compass to facilitate easy data handling.

Git will be employed for version control, using GitHub or GitLab to host repositories and manage collaborative development. Modern web browsers like Google Chrome will be utilized, with built-in developer tools and extensions like React Developer Tools for effective debugging and development.

This environment setup ensures a streamlined development process, allowing for efficient coding, collaboration, and management of the real estate platform.

2.3 REQUIREMENTS TO RUN THE APPLICATION

To run the real estate platform, you need an operating system such as Windows, Linux, or macOS. The application requires Node.js version 14.x or later and MongoDB version 4.x or later, either installed locally or accessed via a cloud service like MongoDB Atlas. Modern web browsers like Google Chrome or Mozilla Firefox are necessary for accessing the platform. For development, a machine with at least 4 GB of RAM and a 2 GHz dual-core processor is recommended, while deployment should be on a server with at least 2 GB of RAM and 1 CPU core.

3. DATABASE ANALYZING, DESIGN AND IMPLEMENTATION

The database for the real estate platform will use MongoDB, a NoSQL database, to handle the diverse and dynamic nature of property listings and user data. The schema design will focus on flexibility, allowing for easy adjustments as requirements evolve. Key collections will include `Users` (for managing buyer, seller, and agent profiles), `Properties` (for storing property details, images, and status), `Listings` (for property availability and pricing), and `Transactions` (for managing sales or rental agreements). The design will incorporate indexing on frequently queried fields like location and price to optimize search performance.

Implementation will involve setting up a MongoDB database, either locally or via a cloud service like MongoDB Atlas, and integrating it with the Node.js back-end using the Mongoose library. Mongoose will provide a schema-based solution to model the application data and handle database operations efficiently. Data validation, relationships between collections, and CRUD[2] (Create, Read, Update, Delete) operations will be implemented to ensure the integrity and accessibility of the data, supporting the platform's functionalities like property search, user management, and transaction processing.

4.PROGRAM's STRUCTURE ANALYZING AND GUI CONSTRUCTING

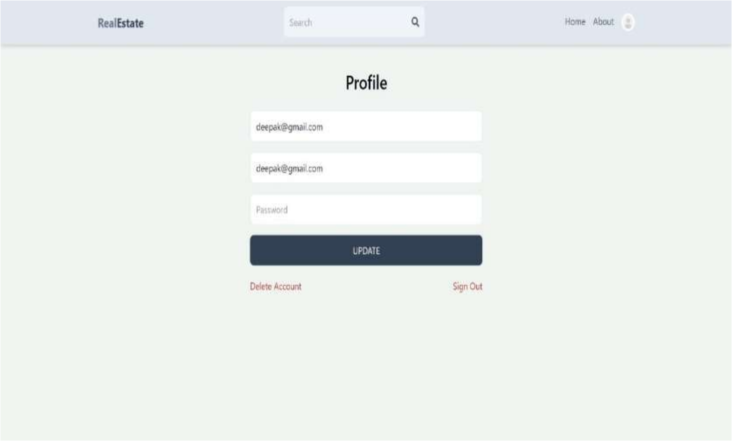
Our real estate application is structured to provide an intuitive and efficient user experience while handling complex data interactions robustly. The back-end is built on a modern web framework that interfaces with a relational database, ensuring data integrity and responsive query performance. Each module corresponds to a core functionality of the real estate domain, including property listings, user management, booking appointments.

GUI CONSTRUCTING:

The graphical user interface of our real estate application is designed with the end-user in mind, aiming for a clean, accessible, and engaging experience. We employed a responsive design framework to ensure the application is accessible across various devices and screen sizes, from desktops to mobile phones.

Navigation is streamlined through a well-organized[1] layout, with clear, actionable items positioned to minimize user interactions for performing common tasks.

PROJECT SNAPSHOTS:

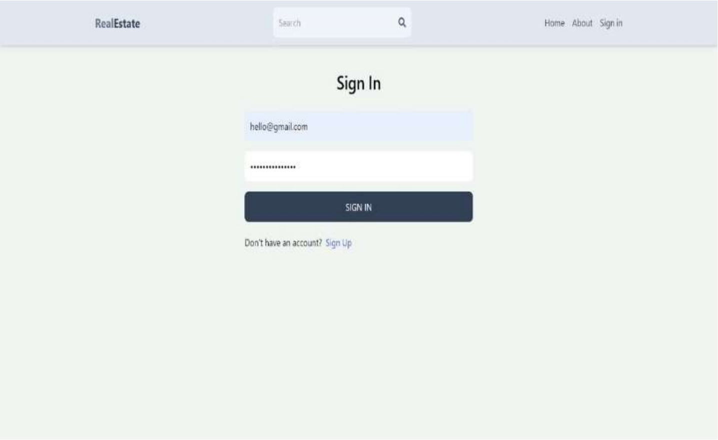


Profile

deepak@gmail.com

deepak@gmail.com

Password

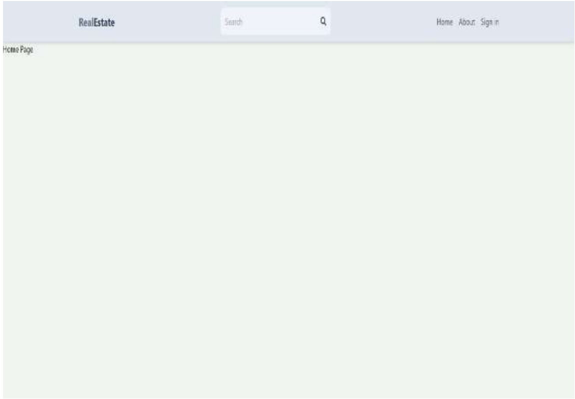


Sign In

hello@gmail.com

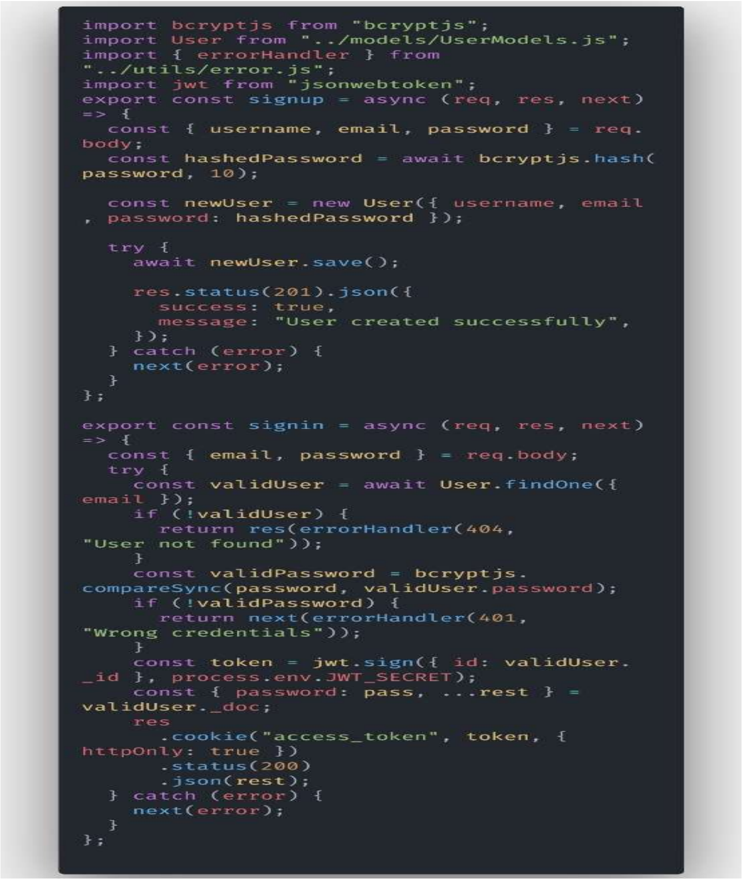
•••••••••••••••

Don't have an account? Sign Up

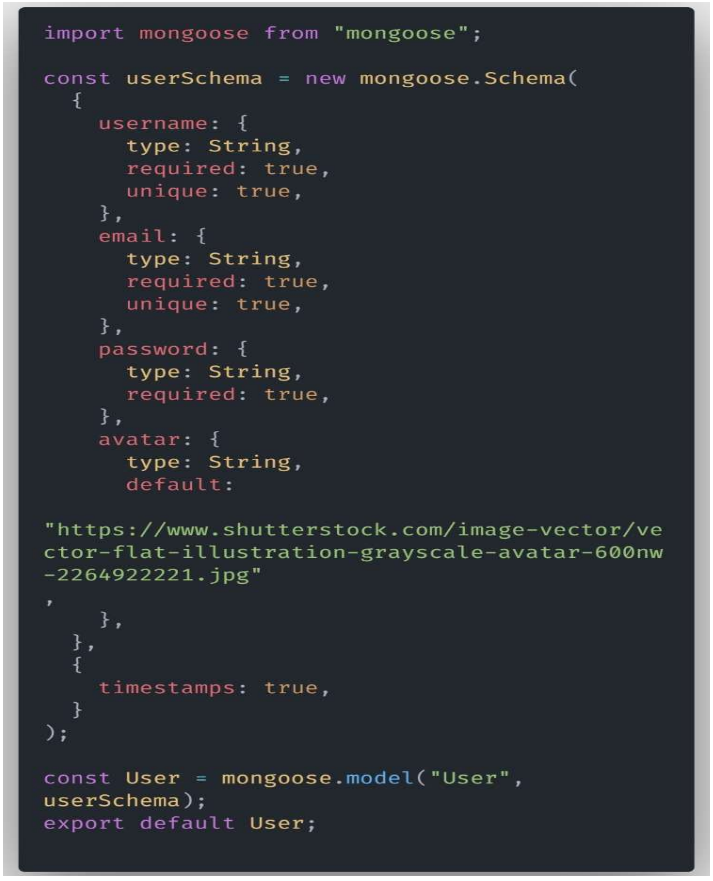


Home Page

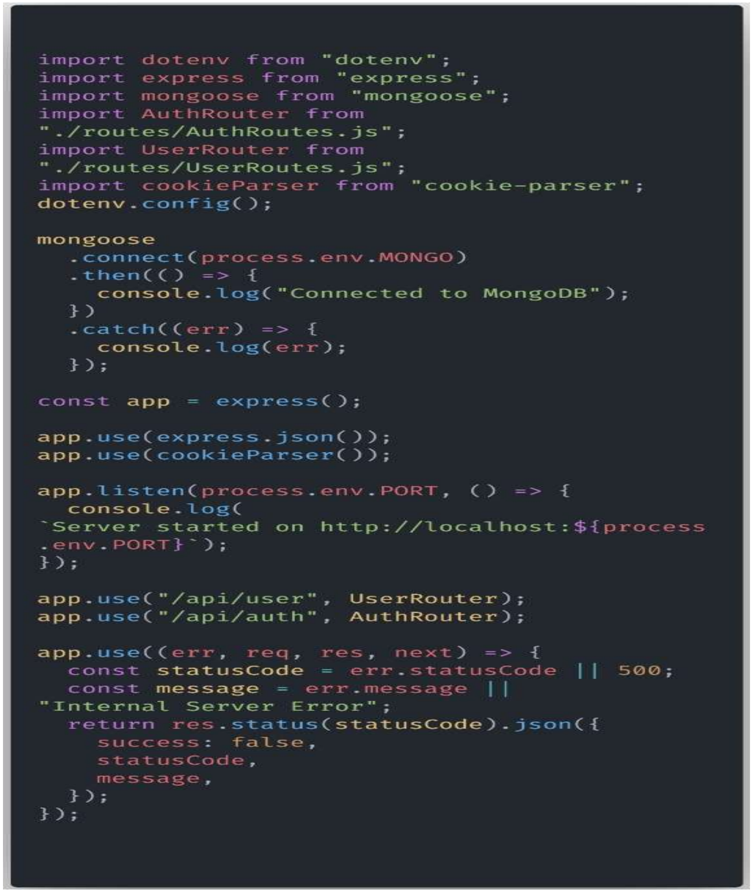
5. CODE-IMPLEMENTATION AND DATABASE CONNECTIONS



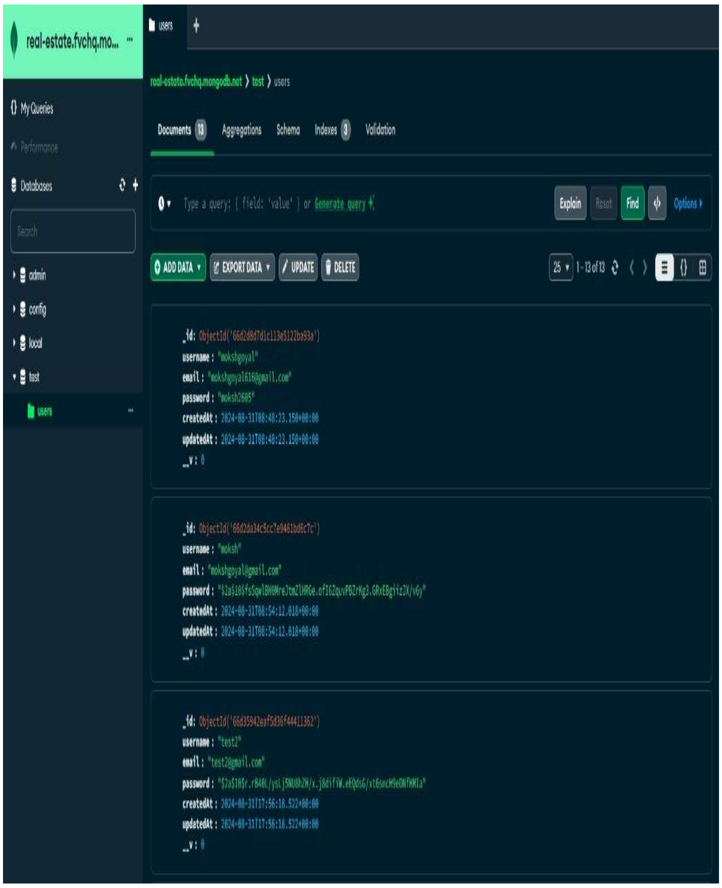
```
import bcryptjs from "bcryptjs";
import User from "../models/UserModels.js";
import { errorHandler } from
"../utils/error.js";
import jwt from "jsonwebtoken";
export const signup = async (req, res, next)
=> {
  const { username, email, password } = req.
body;
  const hashedPassword = await bcryptjs.hash(
password, 10);

  const newUser = new User({ username, email
, password: hashedPassword });

  try {
    await newUser.save();

    res.status(201).json({
      success: true,
      message: "User created successfully",
    });
  } catch (error) {
    next(error);
  }
};

export const signin = async (req, res, next)
=> {
  const { email, password } = req.body;
  try {
    const validUser = await User.findOne({
email });
    if (!validUser) {
      return res(errorHandler(404,
"User not found"));
    }
    const validPassword = bcryptjs.
compareSync(password, validUser.password);
    if (!validPassword) {
      return next(errorHandler(401,
"Wrong credentials"));
    }
    const token = jwt.sign({ id: validUser.
_id }, process.env.JWT_SECRET);
    const { password: pass, ...rest } =
validUser._doc;
    res
      .cookie("access_token", token, {
httpOnly: true })
      .status(200)
      .json(rest);
  } catch (error) {
    next(error);
  }
};
```



```
import mongoose from "mongoose";

const userSchema = new mongoose.Schema(
  {
    username: {
      type: String,
      required: true,
      unique: true,
    },
    email: {
      type: String,
      required: true,
      unique: true,
    },
    password: {
      type: String,
      required: true,
    },
    avatar: {
      type: String,
      default:

"https://www.shutterstock.com/image-vector/ve
ctor-flat-illustration-grayscale-avatar-600nw
-2264922221.jpg"
,
    },
  },
  {
    timestamps: true,
  }
);

const User = mongoose.model("User",
userSchema);
export default User;
```



```
import dotenv from "dotenv";
import express from "express";
import mongoose from "mongoose";
import AuthRouter from
"./routes/AuthRoutes.js";
import UserRouter from
"./routes/UserRoutes.js";
import cookieParser from "cookie-parser";
dotenv.config();

mongoose
  .connect(process.env.MONGO)
  .then(() => {
    console.log("Connected to MongoDB");
  })
  .catch((err) => {
    console.log(err);
  });

const app = express();

app.use(express.json());
app.use(cookieParser());

app.listen(process.env.PORT, () => {
  console.log(
`Server started on http://localhost:${process
.env.PORT}`);
});

app.use("/api/user", UserRouter);
app.use("/api/auth", AuthRouter);

app.use((err, req, res, next) => {
  const statusCode = err.statusCode || 500;
  const message = err.message ||
"Internal Server Error";
  return res.status(statusCode).json({
    success: false,
    statusCode,
    message,
  });
});
```



```
_id: ObjectId('66d2d8d7d1c113e5122ba93a')
username: "mokshgoyal"
email: "mokshgoyal616@gmail.com"
password: "moksh2605"
createdAt: 2024-08-31T08:48:23.150+00:00
updatedAt: 2024-08-31T08:48:23.150+00:00
__v: 0

_id: ObjectId('66d2da34c5cc7e9461bd6c7c')
username: "moksh"
email: "mokshgoyal@gmail.com"
password: "$2a$10$fs5qwlBH0MreJtmZlHRGe.ofI6ZquvPBZrKg3.GRxEBgiizJX/v6y"
createdAt: 2024-08-31T08:54:12.018+00:00
updatedAt: 2024-08-31T08:54:12.018+00:00
__v: 0

_id: ObjectId('66d35942eaf5d36f44411362')
username: "test2"
email: "test2@gmail.com"
password: "$2a$10$r.rB40L/ysLj5NU8hZH/x.j8difiW.eEQdsG/xt6sncM9eDNfMMIa"
createdAt: 2024-08-31T17:56:18.522+00:00
updatedAt: 2024-08-31T17:56:18.522+00:00
__v: 0
```

6. LIMITATIONS

1. Data Accuracy and Completeness

- Description: The accuracy and completeness of property listings are dependent on the data provided by external agents or users. Inaccurate or incomplete data entries can lead to misleading information being displayed on the platform, which could affect user decisions and platform credibility.

- Impact: Users may experience dissatisfaction due to discrepancies between the listed and actual property features, potentially leading to reduced trust in the platform.[3]

1. Scalability Concerns

- Description: As the platform grows, increased data volume could strain the current database and server infrastructure, leading to slower response times and lower overall system performance.

- Impact: This may affect user experience during peak usage times and limit the ability of the platform to expand to new markets without significant upgrades to the infrastructure.

2. Regulatory and Compliance Requirements

- Description: The real estate sector is highly regulated, and compliance with all relevant laws and regulations is mandatory. Changes in legislation could necessitate substantial modifications to the platform to maintain compliance.
- Impact: Failing to comply with new regulations can result in legal challenges, fines, and a damaged reputation.

7. CONCLUSION

As we reach the first milestone of our real estate platform development, we take this opportunity to reflect on our journey so far and set our sights on the next phases of development. The initial quarter of the project has been foundational, laying the technical and strategic groundwork necessary to ensure a robust and scalable solution that will revolutionize the way real estate transactions are conducted.

8. FUTURE SCOPE

1. Technological Advancements:

- Adaptation to New Technologies: The project has the potential to integrate new technologies as they emerge, such as artificial intelligence, machine learning, and blockchain. This would enhance the project's capabilities in data analysis, security, and automation.

2. Scalability:

- Handling Increased Load: The project's architecture can be designed to scale both vertically and horizontally to handle increased user load and data processing demands.
- Expansion to New Markets or Regions: There is scope to expand the service or product to new geographic regions, adapting to local requirements and regulations.

3. User Experience Enhancements:

- Personalization: By leveraging data analytics, the project can offer more personalized experiences to users, improving engagement and satisfaction.
- Accessibility Improvements: Enhancing the accessibility of the project to cater to a wider range of users, including those with disabilities.

References

[1] Learn web development. (n.d.). MDN Web Docs. Retrieved September 19, 2024, from https://developer.mozilla.org/en-US/docs/Learn

[2] What is fullstack JavaScript. (n.d.). W3schools.com. Retrieved September 23, 2024, from https://www.w3schools.com/whatis/whatis_fullstack_js.asp

[3] MERN stack. (2019, October 11). GeeksforGeeks. https://www.geeksforgeeks.org/mern-stack/

[4] Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack Overflow. Retrieved September 23, 2024, from https://stackoverflow.com/