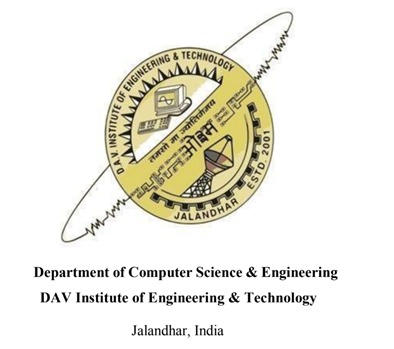
**GHAR KI TALAASH**

**WEBSITE FOR PG HUNTING**

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

SUMMER PROJECT

Duration (from JUNE 2K24 - JULY 2K24 )

SUBMITTED BY Name :- Rimpi

Branch :- Btech ECE

Roll No :- 39/23

Univ. Roll No :-2301765

# **Abstract**

This project is a PG hunting website built using HTML, CSS, JavaScript, React, and Firebase. It helps users easily find and list paying guest accommodations across India. The platform includes separate panels for users, PG owners, and admins, along with features like filtered search, login/signup, and real-time updates. The aim is to make the process of finding PGs simpler, faster, and more organized through a clean and user-friendly website.

**Acknowledgments**

With deep gratitude, I extend my heartfelt thanks to Mrs Neeru Malhotra, Head of the Department Electronics & Communication Engineering, for their unwavering support, inspiration, and for providing the platform to undertake this project.

I am profoundly thankful to the entire team at O7 Services for their exceptional mentorship and insightful training, which greatly enriched my learning experience and contributed significantly to the successful completion of this project.

I also express my sincere appreciation to my faculty, mentors, and peers for their encouragement, guidance, and support throughout this journey.

**Table of contents**

ABSTRACT

ACKNOWLEDGMENTS

TABLE OF CONTENTS

COMPANY PROFILE

**COMPANY PROFILE**

**CHAPTER-1**

# **1.1 Introduction**

In today’s fast-paced world, finding safe and budget-friendly accommodation is a common challenge for students and working professionals, especially in cities. The traditional methods of PG (Paying Guest) hunting are often unorganized, time-consuming, and unreliable. As a team, we identified this gap and decided to develop a PG Hunting Website to make the process simpler, faster, and more transparent.

Our project aims to create a user-friendly platform where PG seekers can easily search for accommodations based on location, budget, amenities, food availability, and gender preferences. At the same time, it allows PG owners to list and manage their properties efficiently.

The website is developed using HTML, CSS, JavaScript, React, and Firebase, and includes separate panels for users, PG owners, and admins. The focus is on delivering a responsive, secure, and functional experience for all users. This project not only enhances digital accessibility in the rental space but also provides a real-world solution to a growing urban need.

# **1.2 Problem Definition**

There is a need for a centralized digital platform that simplifies the PG hunting process by offering reliable listings, user-friendly filters, and direct interaction between owners and tenants. This project aims to address these challenges by developing a responsive, easy-to-use, and efficient web application for PG searching and listing.

# **1.3 Objectives of the Project**

* To simplify the PG search process for students and working professionals by providing an organized and user-friendly platform.
* To allow PG owners to list and manage their properties efficiently with relevant details like rent, amenities, location, and contact info.
* To implement filtered search options (e.g., location, budget, food availability, gender preference) to help users find suitable accommodations quickly.
* To create separate panels for users, PG owners, and admin for better role-based access and management.
* To ensure real-time updates and communication between PG seekers and owners using a dynamic and responsive interface.
* To develop a fully responsive website that works smoothly on mobile, tablet, and desktop devices.
* To use modern web technologies like HTML, CSS, JavaScript, React, and Firebase for a fast, scalable, and secure web application.

# 

**CHAPTER 2**

# **2.1 INTRODUCTION TO LANGUAGES**

Description of each software/technology used in the project:

**1.HTML [Hyper Text Markup Language]**

HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It defines elements like headings, paragraphs, links, images, and more using tags. Browsers read HTML files to display websites visually to users.HTML forms the backbone of all web pages and works with CSS and JavaScript for styling and interactivity.



HTML offers several useful features that make it essential for web development. It uses a tag-based structure that is easy to learn and apply, especially for beginners. It supports embedding multimedia like images, videos, and audio files, and it allows for hyperlinking between pages, creating smooth navigation experiences. HTML also provides form elements, enabling websites to collect user input through contact forms, search bars, and more. Additionally, semantic HTML elements help define the purpose of different sections of a webpage, improving both accessibility and SEO (Search Engine Optimization).

One of the biggest advantages of HTML is its simplicity and universal support across all modern browsers. It is free to use, fast-loading, and SEO-friendly, which helps websites rank better on search engines. Since it's the foundation of web development, it integrates easily with CSS for styling and JavaScript for adding interactive behavior. However, HTML has some limitations. It is primarily used for static content and cannot manage dynamic functionality like data processing or user authentication on its own. It also relies on external tools for advanced styling and scripting. Without a content management system or server-side programming, updating multiple HTML pages can be time-consuming and inefficient.

For our **PG hunting website[Ghar ki Talaash]**, HTML plays a crucial role. It helps to structure the entire website, from the homepage and PG listings to individual landlord or tenant profiles. One can use HTML to display detailed information about each PG, such as photos, rent, amenities, availability, and location. Interactive forms for booking inquiries, contact details, or user registration can also be built using HTML. Additionally, HTML makes it easy to embed maps for location references and videos or images to showcase property interiors. Navigation across different cities, categories, and service pages can be managed efficiently through hyperlinks.

In short, HTML is the essential first step in building our PG hunting website. It lays the groundwork upon which design (CSS) and functionality (JavaScript, back-end programming) can be added to create a complete, user-friendly platform.

**2. CSS [CASCADING STYLE SHEETS]**

CSS (Cascading Style Sheets) is the language used to style and visually design web pages. While HTML handles the structure and JavaScript manages interactivity, CSS controls the layout, colors, fonts, spacing, and responsiveness of a website. It separates content from design, making it easier to manage and update the visual aspects of a site without altering the HTML content itself.

****

CSS provides powerful features that allow developers to create clean, visually consistent, and responsive designs across all screen sizes. We can define style rules once and apply them across multiple pages or elements using classes and IDs. CSS supports media queries, enabling different styles for different devices like mobiles, tablets, and desktops. It also offers advanced layout systems such as Flexbox and Grid, which help organize content in intuitive, flexible ways. With CSS animations and transitions, developers can add smooth visual effects that enhance user experience.

**Advantages of CSS:**

**Separation of Concerns:** Keeps content (HTML) and design (CSS) separate, making code cleaner and easier to maintain.  
**Consistency:** Styles can be reused across multiple pages for a uniform look and feel.  
**Responsive Design:** Media queries make it easy to adapt layouts for different screen sizes and devices.  
**Improved Load Speed:** CSS is lightweight and can be cached by browsers for faster performance.  
**Customizability:** Easy to tweak colors, fonts, spacing, and layout without touching the HTML.

**Disadvantages of CSS:**

**Browser Compatibility:** Some styles may not behave the same across all browsers, requiring extra adjustments.  
**Complexity with Large Projects:** Managing large stylesheets or overrides can become difficult without proper structure.  
**No Logic or Interactivity:** CSS cannot handle dynamic data or user input — it only affects visual appearance.  
**Dependency on HTML Structure:** A change in the HTML layout can sometimes break the CSS styles if not managed carefully.

For our **PG hunting website**, CSS is essential to make the platform visually appealing and user-friendly. We use CSS to design attractive listing cards that display property images, rent details, amenities, and location neatly. It helps create clean layouts for your homepage, filter bars, property grids, and user dashboards. CSS ensures that the website looks great on all devices, whether someone is browsing PGs on a phone or a desktop. It allows you to highlight important elements like "Book Now" buttons or "Verified PG" tags with colors and effects, improving user interaction. Animations can make transitions smoother, such as sliding menus, hover effects on cards, or fading tooltips. Overall, CSS makes your PG website not just functional but visually professional and trustworthy — helping users feel confident while browsing or making decisions.

CSS provides powerful features that allow developers to create clean, visually consistent, and responsive designs across all screen sizes. We can define style rules once and apply them across multiple pages or elements using classes and IDs. CSS supports media queries, enabling different styles for different devices like mobiles, tablets, and desktops. It also offers advanced layout systems such as Flexbox and Grid, which help organize content in intuitive, flexible ways. With CSS animations and transitions, developers can add smooth visual effects that enhance user experience.

**3.JAVASCRIPT**

JavaScript is a powerful, high-level programming language that brings interactivity and dynamic behavior to websites. While HTML structures the content and CSS styles it, JavaScript controls how the website behaves in response to user actions. It runs directly in the browser, making it possible to update content, validate forms, respond to clicks, fetch data from servers, and more — all without reloading the page.

****

JavaScript comes with a wide range of features that make it essential for modern web development. It enables event-driven programming, meaning you can set up interactive responses to user inputs like clicks, scrolls, and keystrokes. JavaScript supports manipulating HTML and CSS in real time through the Document Object Model (DOM), which means developers can create pop-ups, dynamic content updates, sliders, modals, and more. It also allows for asynchronous communication with servers using technologies like AJAX and Fetch API, enabling seamless data loading without refreshing the page.

The advantages of JavaScript are significant. It enhances user experience by making websites faster, more responsive, and more interactive. It works on all major browsers and is supported across devices, from desktops to smartphones. JavaScript also has a huge ecosystem, with countless libraries and frameworks like React, Angular, and Vue.js that accelerate development and add powerful features. Since it runs on the client-side, it reduces the load on the server and improves performance. Additionally, JavaScript is versatile — it can now be used on the server-side as well, thanks to environments like Node.js.

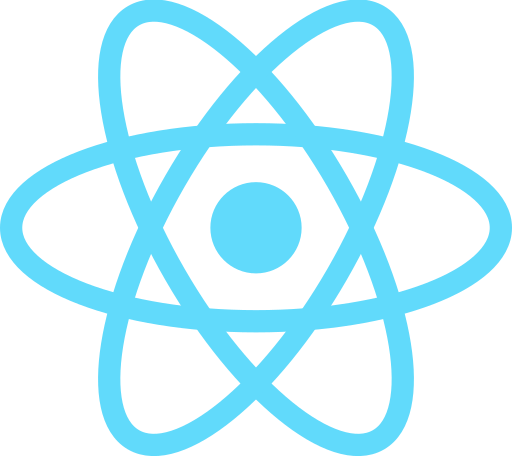
However, JavaScript does come with some disadvantages. Because it runs on the user's browser, its execution can vary slightly between different browsers and devices, requiring thorough testing. It can also be disabled by users, although this is uncommon. Poorly written JavaScript can lead to performance issues or security vulnerabilities like cross-site scripting (XSS). Additionally, JavaScript relies on the browser's processing power, which might affect older or slower devices.

For our **PG hunting website**, JavaScript would significantly enhance both functionality and user experience. We are using it to build dynamic search filters, so users can quickly narrow down PG listings by price, location, gender preference, amenities, or availability. JavaScript can power real-time form validation, alerting users to errors or missing fields before submitting a booking or inquiry. It can be used to create smooth image sliders, interactive maps, and tabs for switching between PG details, reviews, and contact information. With JavaScript, one can also integrate third-party APIs (like Google Maps) to show property locations or fetch real-time data like weather or transit info nearby. It also makes it possible to implement features like user login/logout, favorites/bookmarked listings, and even chat support widgets.

In summary, JavaScript is what makes the website come alive. It turns static pages into interactive platforms, improves user engagement, and enables real-time, intelligent features — making the platform not only functional but also intuitive and modern.

**4. React js**

**React.js** is a powerful JavaScript library developed by Facebook for building dynamic and responsive user interfaces. It allows developers to create interactive web applications using a component-based architecture. Instead of managing entire HTML pages, React breaks the UI into reusable pieces called components, which can be individually built, maintained, and updated. This modular approach simplifies development and improves efficiency, especially for large and complex websites.



One of the key strengths of React is its use of the virtual DOM. Rather than updating the entire page when something changes, React updates only the parts of the interface that have actually changed. This makes the application faster and more efficient, improving the user experience. React also uses a syntax called JSX, which lets developers write HTML-like code directly within JavaScript, making the codebase easier to understand and work with. With built-in support for state and props, developers can manage dynamic data and render content conditionally based on user interaction or real-time input.

React has many advantages. Its component-based structure promotes reusability, making it faster to build and maintain consistent UI elements across the site. The virtual DOM ensures fast rendering performance, while the large ecosystem surrounding React—such as React Router for navigation or Redux for state management—adds powerful capabilities with minimal effort. Server-side rendering is also possible with tools like Next.js, making React-friendly for SEO when set up correctly. Additionally, React has a strong community and excellent developer tools that help with debugging and optimization.

However, React is not without its challenges. It has a steep learning curve, especially for beginners who need to understand concepts like props, state, hooks, and lifecycle methods. The rapid pace of updates in the React ecosystem can make it hard to keep up. Setting up a full React project often requires build tools and configuration, such as Webpack or Vite, which may feel overwhelming at first. Also, React is primarily a client-side technology, so achieving optimal SEO and initial page load speed might require server-side rendering or static generation—extra steps for those who need better search engine visibility. Finally, heavy use of JavaScript can lead to performance issues on lower-end devices if not optimized properly.

React.js would be a powerful tool for building **my PG hunting website**. It allows me to create a fast, dynamic, and modern user experience where users can browse PG listings, apply filters in real-time, and view detailed pages without any full-page reloads. Using React components, I can create reusable cards for PG listings, filter bars, search fields, and user dashboards, saving development time and ensuring consistency. With React Router, I can manage smooth navigation between the homepage, city-based results, PG details, and contact pages. State management enables features like user login, saving favorites, booking requests, and form validation—all of which enhance interactivity and functionality. React also allows me to update data and content dynamically, making my website feel more like an app. This is especially useful for handling large amounts of PG listings, user feedback, or admin management features.

In summary, React.js gives me the tools to build a highly interactive, scalable, and professional PG hunting platform that’s both efficient for development and seamless for users.

**5. GITHUB**

GitHub is a cloud-based platform used for version control and collaboration in software development. It is built around Git, a distributed version control system that tracks changes to code over time. GitHub adds a web-based interface and a range of tools that allow developers to manage code, track progress, collaborate with others, and deploy software more efficiently. It is widely used by individuals, teams, and companies to host and manage software projects of all sizes.

One of the core features of GitHub is its ability to track changes through commits. Each time you or a collaborator makes changes to your code, you can create a commit that records what was changed and why. This history is valuable for understanding project evolution, fixing bugs, or reverting back to previous versions if something goes wrong. GitHub also supports branching, which allows you to work on new features or test ideas in isolation from your main codebase. Once the feature is ready, it can be merged back safely without disrupting the main version of your site.

GitHub also provides powerful collaboration tools. You can work with others on the same project, review code through pull requests, and use issues and project boards to track bugs, feature requests, and development tasks. This makes it ideal not just for individual development but also for team projects. GitHub repositories can be public or private, allowing you to control access to your code. You can also integrate GitHub with other tools like CI/CD services, static site generators, and deployment platforms like Vercel or Netlify, making it easier to automate testing and publish updates.

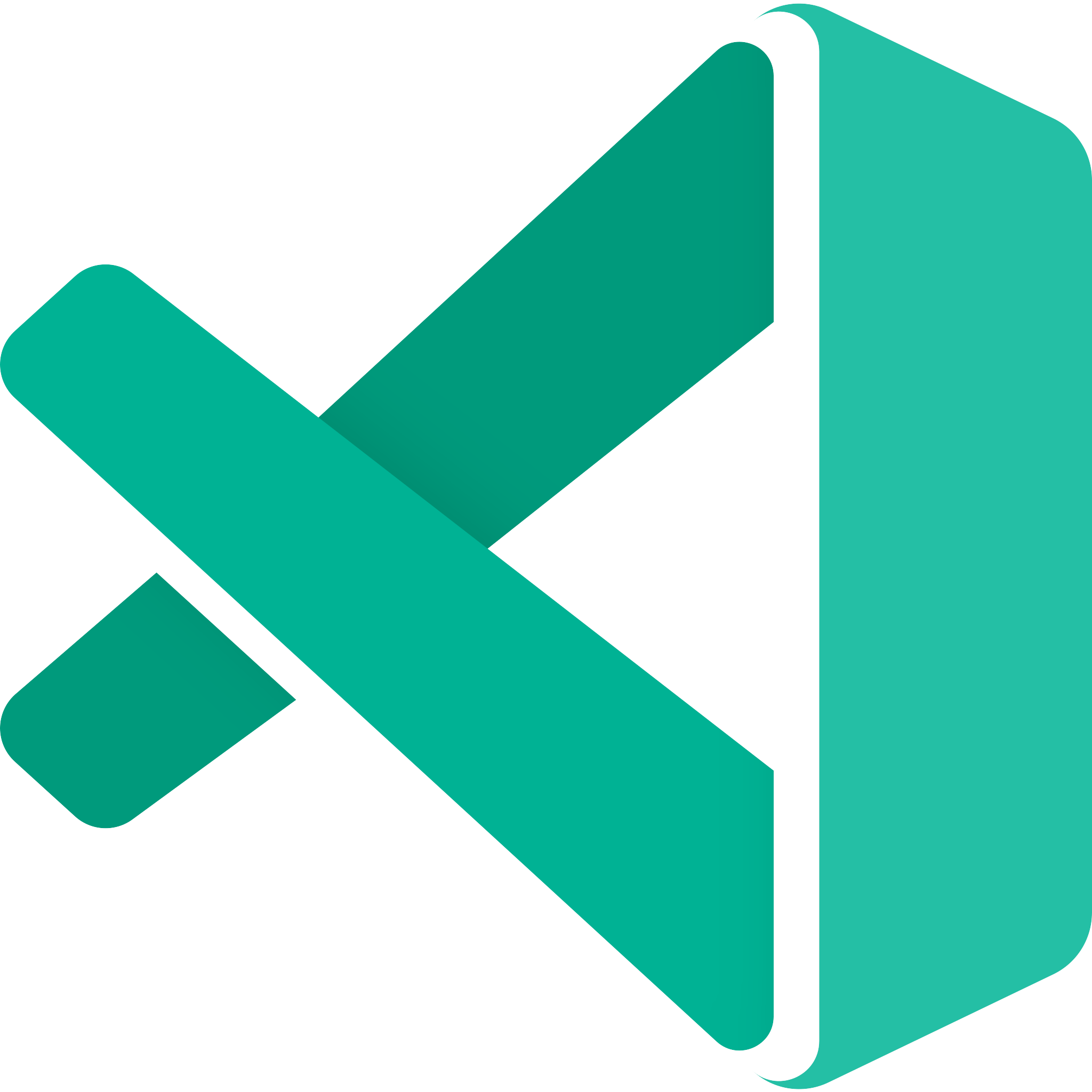
GitHub has several key advantages. It improves code management, allowing you to safely track every change and collaborate with others efficiently. It provides backups of your work in the cloud, reducing the risk of losing your code. It also encourages collaboration and transparency, helping team members review, discuss, and improve code quality. GitHub also integrates with thousands of developer tools and services, making it an essential hub in modern web development. However, there are some disadvantages. For beginners, GitHub (and Git itself) can feel complex at first, especially when using commands like branching, merging, and resolving conflicts. While private repositories are available, free users used to be limited in the past (though GitHub has since lifted many of those restrictions). Additionally, security is important—accidentally pushing sensitive data (like passwords or API keys) to a public repo can be risky if not handled properly.

For my PG hunting website, GitHub is an incredibly useful tool. It allows me to keep a complete history of all the changes I make to my website’s code, so I can easily go back if I break something or want to restore a previous version. If I ever collaborate with a developer, designer, or backend engineer, we can work together without interfering with each other’s changes by using branches and pull requests. I can also use GitHub to manage tasks and bugs through issues and project boards, which helps me stay organized. Most importantly, GitHub allows me to host my code online and even connect it directly to deployment platforms—so any time I update my repository, the changes can be automatically reflected on my live website. It serves as both a safety net and a control center for managing my entire development workflow.

In short, GitHub gives me the tools I need to build, improve, and manage my PG hunting website more effectively, whether I’m working alone or with a team.

**6. VS CODE**

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It is one of the most popular tools used by developers for writing, editing, and managing code across many programming languages. VS Code is lightweight, fast, and highly customizable, making it ideal for web development projects of all sizes—from simple HTML pages to complex applications using modern frameworks like React, Node.js, or Django.



One of the standout features of VS Code is its intelligent code editing. It provides syntax highlighting, code suggestions (IntelliSense), and auto-completion based on the language you’re using. This not only speeds up your coding but also helps prevent errors. VS Code also supports real-time debugging, which lets you run your code and catch bugs directly within the editor. Additionally, it has built-in support for Git integration, so you can manage version control, commit changes, and push to GitHub without leaving the editor.

VS Code’s extension marketplace is another powerful feature. You can add thousands of extensions to enhance functionality—such as live server previews, code formatters, linters, themes, and support for frameworks like React, Angular, Vue, Python, and more. The editor is also highly customizable, allowing you to set keyboard shortcuts, themes, layouts, and workspace settings to fit your preferences. VS Code works across platforms (Windows, macOS, and Linux), ensuring flexibility no matter what operating system you’re using.

VS Code offers many advantages. It is lightweight yet powerful, supporting almost every programming language and tool you need for modern web development. Its user interface is clean and easy to navigate, even for beginners. The live server extension, for example, lets you preview changes to your website instantly in the browser, which is incredibly useful during design and testing. It also includes integrated terminal support, allowing you to run commands, compile code, or install packages without switching applications. Moreover, the built-in Git support makes it easy to manage version control right from your editor. However, there are some downsides. For very large projects or heavy extensions, VS Code can become resource-intensive. Beginners may also feel overwhelmed by the number of features and settings at first. Additionally, while it’s great for writing code, VS Code is not a full integrated development environment (IDE) and may lack advanced tools that some traditional IDEs provide.

For my PG hunting website, VS Code is the ideal workspace. It allows me to write and organize my HTML, CSS, JavaScript, and React code in a structured and efficient way. With the help of extensions, I can preview my PG listing pages live in the browser, test responsiveness, and make quick visual adjustments. I can use the integrated terminal to install packages, run my local development server, or deploy updates to GitHub or hosting platforms. VS Code also helps me stay productive by highlighting errors, suggesting fixes, and keeping my project files well-organized. Its Git integration is especially useful when I’m syncing changes to GitHub, managing branches, or collaborating with others. Whether I’m designing layouts, writing components, or debugging functionality, VS Code provides a smooth, powerful environment to build and improve every part of my website.

In summary, VS Code gives me a complete, professional-grade toolkit to efficiently develop, manage, and grow my PG hunting website, all from one editor.

**8. Browser Developer Tools**

Every modern browser (like Google Chrome) includes built-in Developer Tools. These were used to:

\* Inspect HTML and CSS to debug layout issues.

\* Monitor console errors and network requests.

\* Test website responsiveness on different screen sizes.

\* Optimize performance and user experience.

# **CHAPTER 3**

# **SOFTWARE AND HARDWARE REQUIREMENTS**

### ***1. Software Requirements***

A. Frontend Development

1. HTML5 – for structuring the website.
2. CSS3 – for styling and layout.
3. JavaScript – for interactivity and logic.
4. React.js – to build dynamic, component-based user interfaces.
5. Visual Studio Code – as the primary code editor.
6. Node.js and npm – for running the development environment and managing packages.
7. Google Chrome or Mozilla Firefox – for testing and debugging in modern browsers.
8. Git and GitHub – for version control and code hosting.

**B. Backend Development (Optional)**

1. Node.js with Express.js – for building the server-side application (if needed).
2. MongoDB / MySQL / PostgreSQL – as the database system.
3. Mongoose / Sequelize – for database modeling and communication.

**C. Deployment and Hosting**

1. Vercel or Netlify – to host the frontend (React) website.
2. GitHub Pages – an alternative for hosting static websites.
3. Render / Railway / Heroku – for hosting the backend (if applicable).
4. MongoDB Atlas or Firebase – for cloud-based database services.

### ***2. Hardware Requirements***

**A. For Development (Developer's System)**

1. Processor: Minimum Intel i3 or AMD Ryzen 3; Recommended Intel i5/i7 or Ryzen 5/7.
2. RAM: Minimum 4 GB; Recommended 8 GB or more.
3. Storage: At least 10 GB of free space; SSD preferred for performance.
4. Operating System: Windows 10+, macOS, or any Linux distribution.

**B. For End Users (Website Visitors)**

1. Device: Any smartphone, tablet, or computer with a modern web browser.
2. Browser: Latest version of Chrome, Firefox, Safari, or Edge.
3. Internet: A stable internet connection for loading content and media efficiently.

## **🔧 Tools Used in the Project**

### 1. Frontend Development

* HTML5 – For structuring web content
* CSS3 – For designing and styling the UI
* JavaScript – For interactivity and logic
* React.js – For creating reusable UI components and building the application

### 2. Development Environment

* Visual Studio Code – Code editor for writing and organizing the project
* Node.js & npm – Required for React setup and dependency management
* Git – For version control
* GitHub – For hosting the source code and collaborating

### 3. Design & Preview Tools

* Figma / Adobe XD – For planning and designing the website layout (optional)
* Live Server / Vite / React Scripts – For running the development preview

### 4. Deployment

* Vercel / Netlify – For hosting the static frontend website
* GitHub Pages – Optional static site deployment alternative

# **CHAPTER 4**

**FEASIBILITY STUDY**

**1)Technical Feasibility**

From a technical perspective, this stack is entirely feasible and well-suited for building a responsive PG (Paying Guest) hunting website. React allows you to develop a modular and interactive user interface, where components like listing cards, search filters, and dashboards can be easily managed and reused. With HTML and CSS as your foundational technologies and Bootstrap layered in, we can quickly achieve a responsive, mobile-friendly layout without the need for advanced CSS frameworks. Bootstrap’s grid system and prebuilt UI components reduce the time needed to create a consistent and polished interface, while still allowing room for customization through your own CSS.

On the backend, Firebase offers a serverless and scalable platform that integrates smoothly with React. Firebase Authentication supports secure user login using email/password or third-party providers like Google. Firestore, the NoSQL database provided by Firebase, is highly flexible and capable of storing PG listings, user profiles, inquiries, and reviews. Real-time syncing of data is a major advantage for features like availability status or live updates. Firebase Storage allows you to handle image uploads efficiently, and Firebase Hosting makes it easy to deploy your website with HTTPS and CDN support—all without managing any servers.

**2)Operational Feasibility**

In terms of operational feasibility, this stack reduces overhead significantly. Firebase removes the need for setting up and maintaining traditional backend infrastructure, making the project easier to manage for small teams or individual developers. One can manage listings, users, and uploaded content through the Firebase Console. Admin features such as dashboard access, PG listing management, and role-based access control can be implemented using Firebase Authentication combined with Firestore permissions.

**3)Financial Feasibility**

From a financial standpoint, this stack is cost-effective, particularly for MVPs or early-stage projects. Firebase’s free tier includes generous limits for authentication, database usage, storage, and hosting, which should be sufficient for initial development and testing phases. Costs remain low as long as data usage and traffic are moderate. Bootstrap further helps by accelerating development and reducing the need for custom UI design work, which can also reduce time and development costs.

On the user experience side, the combination of React and Bootstrap supports a smooth, responsive interface across all devices. Users can easily navigate through listings, apply filters, view PG details, and interact with features like contact forms or booking options. The design can remain clean and functional while still being visually appealing. Optional features like Google Maps integration can enhance location-based searching but are not essential for a working MVP.

However, there are some challenges to consider. Firestore rules must be carefully configured to secure user data and restrict access appropriately. Since React is client-rendered, SEO will be limited unless we use a framework like Next.js in the future. This is not a major concern if the platform is primarily accessed by logged-in users. Also, if users can post their own PG listings, some form of moderation or verification will be necessary to maintain trust and quality. As the platform scales, Firebase costs may grow, especially in terms of database reads and storage, so query and data structure optimization is important from the start.

In conclusion, using HTML, CSS, JavaScript, Bootstrap, React, and Firebase to build a responsive PG hunting website is both practical and efficient. This stack supports rapid development, real-time features, scalable architecture, and a good user experience without the need for heavy infrastructure or high initial costs. It is well-suited for startups, student projects, or commercial MVPs looking to launch quickly and scale gradually.

**CHAPTER 5**

**SYSTEM ANALYSIS**

In today’s fast-paced urban landscape, students and professionals frequently migrate to new cities, often encountering significant challenges in finding suitable paying guest (PG) accommodations. The traditional approaches ranging from unreliable word-of-mouth recommendations to inefficient social media listings and broker-based solutions are riddled with issues such as lack of verified information, limited availability, and absence of real-time updates. Recognizing these gaps, our proposed system aims to revolutionize the PG-hunting experience by introducing a modern, responsive web application that is seamless, intuitive, and data-driven called “GHAR KI TALAASH”

This system is designed as a fully responsive web platform built using React for the frontend, styled elegantly with Bootstrap and custom CSS, and powered by Firebase’s robust backend services. The application allows users to browse a curated list of PG accommodations filtered by location, budget, and amenities. With Firebase Authentication, users can securely register and log in, while PG owners or administrators can easily manage listings through a personalized dashboard. Firestore ensures real-time data access and synchronization, enabling dynamic updates to availability or pricing. Meanwhile, Firebase Storage handles high-quality image uploads for each listing, and Firebase Hosting ensures secure, fast, and globally accessible deployment.

Compared to the fragmented and outdated methods currently in use, this platform stands out by offering an integrated experience that bridges the gap between PG seekers and providers. It eliminates the inefficiencies of manual listing updates and introduces a more reliable and user-friendly approach to accommodation discovery. Technically, the system leverages React’s component-driven structure for modular development, while Bootstrap ensures mobile-first design with responsive grids and prebuilt UI elements, accelerating development without compromising design quality.

Operationally, the system is lightweight and highly maintainable, thanks to Firebase’s serverless infrastructure. It significantly reduces backend management overhead and allows the development team to focus on enhancing user features rather than maintaining servers. From a financial standpoint, the system is cost-effective, especially during the initial phase, as Firebase offers a generous free tier covering authentication, database usage, hosting, and storage. This makes the platform an ideal choice for startups, student ventures, or early-stage MVPs with limited budgets but high growth potential.

Functionally, the system is rich in features: users can create accounts, search and filter PGs by location or price, view detailed information along with photos, and send inquiries or booking requests. PG owners can list properties, upload images, and manage availability—all from a streamlined dashboard. Non-functional requirements such as responsiveness, performance, security, and scalability are addressed through the use of modern web technologies and cloud infrastructure.

Architecturally, the application follows a clean separation of concerns. The frontend handles user interaction and UI logic using React, while Firebase serves as the backend platform for all services—authentication, data storage, image handling, and hosting. This architecture supports fast development, real-time functionality, and easy deployment. Users of the system fall into three primary categories: PG seekers, property owners/admins, and optionally, a system administrator who may oversee content moderation and platform governance.

However, certain limitations remain. React’s client-side rendering may pose SEO challenges, which can be addressed in future iterations by migrating to server-side rendering frameworks like Next.js. Additionally, without integrated messaging, user-to-owner communication would depend on external means (e.g., phone or email). Moderation of user-submitted listings is also a consideration to maintain content quality and trustworthiness. Nonetheless, these limitations are manageable and do not hinder the core functionality of the platform.

Looking ahead, the system has strong potential for enhancement. Future updates could include integration of Google Maps for geolocation-based browsing, a review and rating system for listings, notification alerts for new PGs, and secure payment gateways for online bookings. These features would further elevate the user experience and bring the platform closer to a comprehensive PG management ecosystem.

In conclusion, the system analysis confirms that developing a responsive PG hunting website using React and Firebase—complemented by HTML, CSS, JavaScript, and Bootstrap—is not only feasible but highly efficient, scalable, and impactful. It addresses a real-world need with a modern technological solution and holds significant promise for future growth and innovation.

**CHAPTER 6**

**SCREENSHOTS**

**CHAPTER 7**

**TESTING**

Testing plays a critical role in the software development life cycle, ensuring that the final product is reliable, functional, and user-friendly. For our responsive PG hunting website developed using HTML, CSS, JavaScript, Bootstrap, React, and Firebase, a structured and comprehensive testing process was conducted to validate all functionalities and performance aspects across different devices and use cases.

#### **7.1. Types of Testing Performed**

1. Unit Testing:  
 Individual components of the React application were tested in isolation to verify that each performs its specific function correctly. This included testing components such as the PG listing cards, search and filter modules, user login forms, and the admin dashboard. React’s state and props handling were rigorously verified using mock data to ensure predictable outputs.

2. Integration Testing:  
 Once individual components were validated, integration testing was performed to ensure they worked correctly together. This included testing the interaction between the frontend and Firebase services, such as data retrieval from Firestore, image uploads to Firebase Storage, and authentication flows via Firebase Auth.

3. Functional Testing:  
 All features outlined in the system requirements were manually tested against expected behaviors. This included user registration and login, adding and editing PG listings, image uploads, search filters, mobile responsiveness, and navigation. Each function was tested using valid and invalid input to ensure appropriate responses.

4. Cross-Browser Testing:  
 To ensure consistency across platforms, the website was tested on multiple web browsers including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Compatibility issues were identified and resolved to maintain consistent user experience.

5. Responsive and Mobile Testing:  
 Given that the platform is designed to be responsive, extensive testing was done across different screen sizes and devices (desktop, tablet, smartphone). Bootstrap’s responsive grid and media queries were evaluated to confirm that layout and content adjusted appropriately across resolutions.

6. Performance Testing:  
 The loading speed and responsiveness of the application were evaluated, especially for pages with multiple images and data-intensive content. Firebase Hosting ensured efficient content delivery, while image sizes were optimized to reduce load times.

7. Security Testing:  
 Authentication and database access were tested against unauthorized access. Firebase Security Rules were tested to ensure that users could only access or modify data relevant to their roles. Input validation was enforced in forms to prevent data injection or malformed entries.

#### **7.2**. **Bug Tracking and Resolution**

All discovered bugs and issues were documented, categorized by severity, and resolved accordingly. Common issues included form validation errors, mobile layout shifts, and asynchronous data fetching delays. Tools like the browser console and Firebase’s debugging logs were used to trace and fix issues efficiently.

**7.3. Final Outcome**

Upon completion of all testing phases, the website proved to be stable, functional, and user-friendly. It met all functional requirements and performed consistently across devices and browsers. The responsive design worked flawlessly, and Firebase integration delivered secure, real-time data interaction. The system is now ready for deployment and real-world use.

**CHAPTER 8**

**SUMMARY AND CONCLUSION**

**summary**

This project report presented the design, development, and implementation of a responsive web-based platform for PG (Paying Guest) accommodation hunting. The website was built using a modern web development stack comprising HTML, CSS, JavaScript, Bootstrap for styling and layout, React for frontend component management, and Firebase for backend services such as authentication, real-time database, storage, and hosting.

The primary goal of the system was to provide a centralized, user-friendly platform where users can easily search for PG accommodations based on their preferences such as location, budget, and amenities, while also allowing property owners to manage their listings. The website features a responsive user interface, secure user authentication, real-time listing management, and efficient image handling. Testing was conducted across multiple devices and browsers to ensure reliability, responsiveness, and functional correctness. Various testing strategies including unit testing, integration testing, performance testing, were applied to validate the system.

Through this project, we aimed to address the inefficiencies of traditional PG-hunting methods and offer a more reliable, scalable, and accessible digital solution. The application stands out by integrating real-time capabilities with a clean and intuitive design, ensuring a smooth experience for both PG seekers and owners.

### **Conclusion**

The development of the responsive PG hunting website successfully demonstrates how modern web technologies can be used to solve real-world problems. By leveraging the strengths of React for dynamic UI development and Firebase for serverless backend services, the platform achieves a high level of functionality with minimal infrastructure complexity.

The project fulfills its objectives by providing a robust solution that simplifies the process of finding and managing PG accommodations. It ensures accessibility across devices, secures user data through authentication and database rules, and offers a scalable foundation for future enhancements such as maps integration, reviews, and online payments.

In conclusion, the website is not only technically sound but also practical in its application. It bridges the gap between technology and everyday housing needs, setting a strong foundation for potential expansion into a full-fledged PG management and discovery ecosystem. The project highlights the effectiveness of full-stack web development and cloud services in building efficient, real-time, and user-centric platforms.