Home Revive

A Project Report
Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Naksha Ramesh Shetty
TIT2324073

Under the esteemed guidance of
Mrs. Sudha Bhagavatheeswaran

DEPARTMENT OF INFORMATION TECHNOLOGY
SIES COLLEGE OF ARTS,SCIENCE &
COMMERCE(AUTONOMOUS)
SION(W),MUMBAI, 400022
MAHARASHTRA
2023

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

Rollno: TIT2324073

- Name of the Student – Naksha Ramesh Shetty
- Title of the Project - Home Revive
- Name of the Guide : Sudha Bhagavatheeswaran
- Teaching experience of the Guide 20 yrs
- Is this your first submission? Yes

Signature of the Student	Signature of the Guide
Naksha Ramesh Shetty	SudhaBhagavatheeswaran
Date: 22nd July 2023	Date: 22nd July 2023

Signature of the Coordinator
Sudha Bhagavatheeswaran
Date:

SIES COLLEGE OF ARTS, SCIENCE & COMMERCE(AUTONOMOUS)
(Affiliated to University of Mumbai)
SION(W), MUMBAI-400022

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Home Revive"**, is bonafide work of **Naksha Ramesh Shetty** bearing Seat.No: **TIT2324073** submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai.

Internal Guide **Coordinator**

SudhaBhagavatheeswaran **SudhaBhagavatheeswaran**

External Examiner

Date: **CollegeSeal**

ABSTRACT

This website HomeRevive will help you to fulfill all your dreams.Our main goal is to ensure our customers are happy by turning their dreams into reality.We all imagine about having a home with all the little touches that makes it uniquely ours.Now a days every persons taste or choice of design is completely different some are more likely to like modern look for there interior and some like the touch of vintage look.Those little details will turn the place into home.At the end of the day we all come back to our comfort zone that is home.Our website will have a login page,design outlet,contractor details,before and after pictures,FAQ,price filters and additional feature where the customer can buy the furniture .After all this steps costomer needs to make a booking.All the designs for different categories like living room, kitchen, bedroom, bathroom etc.After all the renovation we need furnitures to complete the picture so we have additional feature in our website were you can buy the furniture according to the theme.

Customer just have to select the design for the category or if they want to personalize they can have a conversation with constructor and architecture sharing there ideas and making it happen in real and having the accessibility to buy furniture makes the home more luxurious.This website will make your life easy by having all the requirement in one website like desings, contractor, architecture, furniture.

ACKNOWLEDGEMENT

I owe the successful completion of this project to our parents and well-wishers. Additionally, I extend my sincere gratitude to our guide**, Mrs. Biju Ramesh**, and **Mrs. Sudha Bhagavatheeswaran**, H.O.D. of the **IT Department at SIES College of Arts, Science & Commerce (Autonomous)**, for their continuous guidance and support during the entire project. Before delving into the details, I want to express our heartfelt appreciation for all the individuals who played various roles in our project, offering exceptional support from the project's inception. It is our privilege to convey our sincerest regards to our guide**, Mrs. Sudha Bhagavatheeswaran**, for her valuable insights, capable guidance, unwavering encouragement, wholehearted cooperation, and constructive criticism throughout the project's duration.

Furthermore, I deeply express our sincere thanks to our Head of Department, **Mrs. Sudha Bhagavatheeswaran** , for her unwavering encouragement and support. I also take this opportunity to express our gratitude to all our lecturers who, whether directly or indirectly, contributed to the project. Lastly, I offer our respect and gratitude to our parents, as well as all other family members and friends, for their boundless love and encouragement throughout the project.

DECLARATION

I hereby declare that the project entitled, “**HomeRevive**” has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Naksha Ramesh Shetty

Name and Signature of the Student

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Background 01
1.2 Objectives 01
1.3 Purpose, Scope & Applicability 03
1.3.1 Purpose 03
1.3.2 Scope 03
1.3.3 Applicability 04

2. Chapter: Survey of Technologies

2.1 Front end Technologies 05
2.1.1 HTML (Hyper Text Markup Language) 05
2.1.2 CSS (Cascading Style Sheets) 07
2.1.3 React 09
2.1.4 JavaScript 10
2.2 Back end Technologies 12
2.2.1 MongoDB 12
2.2.2 ASP.NET 14
2.2.3 Python 15
2.2.4 FireBase 17
2.3 Technologies used in the project 19

3. Chapter: Requirements and Analysis

3.1. Problem Definition 20
3.2. Requirements Specification 20
3.3. Planning and Scheduling 21
3.3.1. Planning 21
3.3.2. Scheduling 22
3.4. Software and Hardware 25
3.4.1. Software Requirements 25
3.4.2. Hardware Requirements 25
3.5. Conceptual Models 26
3.5.1. Entity-Relation Diagram 26
3.5.2. Data Flow Diagram 27

4. Chapter: System Design

4.1. Basic Modules 29
4.2. Schema Design 30
4.3. User Interface Designs 33
4.4. Security Issues 41

Chapter 1: Introduction

1.1 Background:

A home renovation website is a online hub for fixing up your house. Its a special place on the internet where you can find all the help you need to make your home better. Earlier when we think of renovating our house we have to do everything from scratch. From finding good architecture who understand what you want, to analyzing the designs and try to make them all fit together .What's set this website apart is its team of experts who are readily available for virtual chat. This experts will carefully listen and will understand what you genuinely need and also give advice on how to make them come to life. After all this it doesn't ends here hunting for new furniture is a big challenge. It becomes a hustle and overwhelming for a person having a website which contains all the customer needs and where you can talk to experts who really get your ideas and help make them real. They show you lots of design options that match what you like and when its time to start, they even help you with furniture that fits perfectly. Imagine all the expectations coming into reality without all the stress, that's what Home Revive does it makes home renovation super easy and fun. Moreover, you can explore a gallery of stunning images showcasing real homes that have undergone remarkable transformation. It's like having all your requirements in one place. The best part? All of this happens without the stress and overwhelm that often come with home renovation. So, Home Revive takes your expectation and turns them into reality, making the whole process super easy and fun. Its like having a one-stop shop for all your home renovation needs.

1.2 Objectives:

The main objectives of HomeRevive are:

Transforming Living Spaces: Our primary aim is to enable individuals to turn their living spaces into havens of comfort and beauty, fulfilling their dreams of an ideal home.

User-Friendly Experience: We prioritize user-friendliness, striving to simplify complex processes and make the renovation journey accessible and enjoyable for our customers.

Streamlining Renovations: We simplify the renovation process by having users log in, select their desired renovation category, and engage in productive conversations with our experts.

Diverse Design and Furniture Options: Home Revive offers a wide range of designs and furniture options, catering to various styles, whether it's a modern look or vintage charm.

Understanding the Essence of Home: We deeply understand that a home is where one finds comfort and happiness, and we aim to create spaces that reflect these feelings.

Connecting with Talented Architects: Our platform connects users with skilled architects who can bring their ideas to life, ensuring that each project is executed with expertise and creativity.

Guidance for Selection: We assist users in choosing the right designs and furniture for different categories, ensuring a fun and enjoyable renovation process.

Galleries : Customer can get to see the photo gallery of home which where renovated by the professionals also they can see the before and after of there previous work.

Payment : once customers have finished finding the design they want and have provided all the necessary details, they can then go to the payment section. In this payment section, they can pay for the design and book the architect or service they need.

Customer Satisfaction: Ultimately, our ultimate goal is to leave our customers fully satisfied with our work, making their dreams of a perfect home a delightful reality.

1.3 Purpose, Scope & Applicability:

1.3.1 Purpose:

The purpose of this website is to make things easier for the customer.As a person decides to renovate the house it is hard to find good architecture ,designs ,furnitures as per your taste.This website will solve that problem by putting everthing in one spot.This offers the special place for renovation.Instead of spending a lot of time looking all over the internet,you can find a lot of architects,tons of designs ,and furniture that fits your style all in this website,no more stressing or searching.This website is your friend when it comes to selecting the whether you want morden look or more classic.Get ready for the exciting journey where everything you need is in one place making things fun and creative.

1.3.2 Scope:

Earlier whenever we think of renovating the house we have to go through all the difficulties by finding good architecture who understands want we want and what we expect to be done, then we have to search for the designs online which is time consuming. After all this we have to hunt for good furniture that suits our designs. First the customer has to login the website by filling all the details the they can select the categories like kitchen, bathroom, living room, bedrrom etc then according to the category there will be a design options with there prices and also a photo gallery in which there will be the pictures of the work that the architecture has done previously The customer can directly contact and have a conversation with the experts about there need and what they expect from them. All the queries of the customer will be solved in FAQ section.THis website also helps to get furniture for your house according to your taste and design of the house. After all the this, customer can procedure to the payment section by adding all the details required to book the architects. Moreover, this website is easy to use and this involves gathering all different range of architectural designs, collaborating with professionals, and selecting suitable furniture. This website emphasizes ease of use, efficiency, and accessibility for homeowners who wants to redo there space. It satisfies users need by simplifying there work by getting all the requirement in one place. Getting them a professional collaborations so they can have conversation and share their ideas and get advice to make their home exactly like they want. Even the contractors work will be shared so that the user can see and have some idea and see progress. Before and after pictures will help the customer to have a view of how there project is going to look. Customer can view variety of designs for different categories. Furniture just completes the house and in general we have to go looking for the furniture but this website makes that job easy by have a feature where the customer can buy furniture right from the website. In the end ,this website wants to make home renovation fun by giving the access designs, experts and furniture options all in one. Making peoples life easy and turning there dreams into real life.

1.3.3 Applicability:

This website is easy to use and this involves gathering all different range of architectural designs,collaborating with professionals, and selecting suitable furniture.This website emphasizes ease of use, efficiency, and accessibilty for homeowers who wants to redo there space.It satisfies users need by simplifing there work by getting all the requirement in one place.Getting them a professional collaborations so they can have conversation and share their ideas and get advice to make their home exactly like they want.Even the contractors work will be shared so that the user can have some ideas and see the progress.Before and after pictures will help the customer to have a view of how their project is going to look.Customer can view variety of designs for different categories.Furniture just completes the house and in general we have to go looking for the furniture but this website makes that job easy by have a feature where the customer can buy furniture right from the website.In the end ,this website wants to make home renovation fun by giving the access to designs,experts and furniture options all in one.Making peoples life easy and turning what they have imagine into something real.

Chapter 2 - Survey of Technologies

2.1 Front end Technologies

2.1.1 HTML (Hyper Text Markup Language)

HTML, known as the HyperText Markup Language, serves as the universally accepted markup language for crafting documents meant to be viewed in web browsers. Its primary function is to define the structure and significance of web content. Frequently, it collaborates with tools like Cascading Style Sheets (CSS) for styling and scripting languages such as JavaScript for interactivity.

When web browsers retrieve HTML documents, whether from a remote web server or local storage, they transform these documents into dynamic multimedia web pages. HTML plays a pivotal role in organizing the content on a web page in a meaningful and structured manner. While it originally included instructions for how content should look, modern web development segregates this aspect into CSS for enhanced flexibility and maintainability.

ADVANTAGES

Universal Compatibility: HTML is supported by all web browsers and is a cross-platform language, ensuring that web content is accessible to a wide audience regardless of the device or browser being used.

Ease of Learning: HTML is relatively easy to learn and understand, making it accessible to beginners in web development. It has a simple and straightforward syntax, making it a great starting point for those new to programming.

Semantic Structure: HTML allows for the creation of structured and semantically meaningful content. Elements like headings, paragraphs, lists, and links provide a clear and organized structure to web documents, improving accessibility and search engine optimization (SEO).

Separation of Content and Presentation: While HTML defines the structure and content of a webpage, it encourages the separation of presentation (CSS) and behavior (JavaScript). This separation enhances code maintainability and flexibility, allowing for easier updates and styling changes.

Extensibility: HTML is extensible through the use of attributes, which can add functionality and interactivity to web elements. Additionally, HTML5 introduced new elements and attributes for multimedia (e.g., <video> and <audio>), graphics (e.g., <canvas>), and more.

DISADVANTAGE

Limited Styling and Presentation Control: HTML primarily focuses on the structure and content of web pages. While it provides basic formatting and layout options, it's not designed for complex styling and presentation. Cascading Style Sheets (CSS) are typically used for detailed control over a webpage's appearance.

Lack of Interactivity: HTML alone doesn't offer advanced interactivity or dynamic behavior. For interactive features like form validation, animations, or real-time updates, developers rely on scripting languages like JavaScript. This can complicate web development by requiring multiple technologies to work together.

Browser Compatibility Issues: Although HTML is a standardized language, different web browsers can interpret it slightly differently. This can lead to cross-browser compatibility issues, where a webpage looks or functions differently in various browsers. Developers often need to write additional code to address these differences.

Limited Multimedia Handling: While HTML has evolved to support multimedia elements like images, audio, and video (with HTML5), it may not provide the level of control or features required for complex multimedia applications. For advanced multimedia functionality, developers may need to use additional technologies or frameworks.

Security Vulnerabilities: HTML itself is not inherently insecure, but it can be used to create web pages with security vulnerabilities if not implemented correctly. For example, poorly sanitized user input in HTML forms can lead to security risks such as cross-site scripting (XSS) attacks.

2.1.2 CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) is a language used to style web pages. It tells web browsers how to present content. CSS defines colors, fonts, spacing, and layout. By separating design from content (usually HTML), it keeps websites organized and easy to update. CSS rules select HTML elements and specify how they should look. Styles can be applied directly in HTML, in external CSS files, or even inline within specific elements. CSS is essential for creating visually appealing, consistent, and responsive web designs. It empowers web developers to control the appearance of web pages, making them attractive and user-friendly across different devices.

ADVANTAGE

Separation of Concerns: CSS allows for a clear separation of content (HTML) and presentation (styling). This separation enhances code organization, maintenance, and collaboration among developers and designers.

Consistency: CSS enables consistent styling across a website. You can define styles for elements once and apply them consistently to multiple pages, ensuring a uniform look and feel.

Flexibility: CSS provides extensive control over the visual presentation of web pages. You can adjust fonts, colors, margins, padding, and more to achieve the desired design.

Responsive Design: CSS supports responsive web design, allowing web pages to adapt to various screen sizes and devices. Media queries and flexible layout techniques make it possible to create mobile-friendly and responsive designs.

Efficiency: By using CSS, you can make global style changes quickly. This efficiency is especially valuable when updating the design or layout of a website with many pages.

DISADVANTAGE

Separation of Concerns: CSS allows for a clear separation of content (HTML) and presentation (styling). This separation enhances code organization, maintenance, and collaboration among developers and designers.

Consistency: CSS enables consistent styling across a website. You can define styles for elements once and apply them consistently to multiple pages, ensuring a uniform look and feel.

Flexibility: CSS provides extensive control over the visual presentation of web pages. You can adjust fonts, colors, margins, padding, and more to achieve the desired design.

Responsive Design: CSS supports responsive web design, allowing web pages to adapt to various screen sizes and devices. Media queries and flexible layout techniques make it possible to create mobile-friendly and responsive designs.

Efficiency: By using CSS, you can make global style changes quickly. This efficiency is especially valuable when updating the design or layout of a website with many pages.

2.1.3 React

React is a free and open-source tool that helps people create user-friendly websites and apps. It's like a set of building blocks for the web. These blocks are called components, and you can use them to build web pages and apps quickly.

React is maintained by a company called Meta (formerly Facebook) and a group of developers from different places. It's mainly used for making web applications that feel smooth and fast, whether you're using a web browser or a mobile device.

React focuses on how things look and work on the screen. It doesn't handle things like how to move between different pages of a website; for that, you'd use other tools like Next.js. So, React is all about making your website or app look great and work smoothly for users.

ADVANTAGE

Reusable Building Blocks: React lets you create reusable pieces of your website or app, like LEGO bricks. You build once, then use those pieces again and again to make different parts of your site.

Fast and Smooth: React makes your website work really fast and smoothly, so when you click or tap on something, it responds quickly. Users love when things happen instantly!

Easy to Change: If you want to change how your website looks or works, React makes it easy. You can make updates without messing up everything else.

Lots of Help: Many developers use React, so there are lots of tutorials, guides, and people who can help you if you get stuck. It's like having a big community of friends who know how to build websites.

Works Everywhere: Whether you're using a computer, tablet, or phone, React makes sure your website looks and works nicely on all of them. It's like having a website that's like a chameleon, adapting to any screen.

DISADVANTAGE

Complex for Beginners: React can be tricky to learn at first. It's like trying to ride a bike without training wheels when you're just starting; it takes some practice.

SEO Challenges: React apps are mainly built in JavaScript, which search engines don't understand as well as regular HTML. This can make it harder for your website to show up in search results.

Many Choices: React gives you lots of options for how to do things, which can be confusing. It's like having too many flavors of ice cream to choose from; sometimes, you're not sure which one is best.

Frequent Updates: React updates often, and that can mean you need to make changes to your code to keep up. It's like having to constantly update your phone's apps.

Can Be Overkill: For small and simple websites, React might be like using a bulldozer to dig a small hole. It's powerful, but sometimes you don't need all that power, and it can make your website slower to load.

2.1.4 JavaScript

JavaScript, in simple words, is a programming language that makes websites and web applications do cool things. It's like the magic wand of the internet.

With JavaScript, you can make web pages interactive. You can create pop-up messages, change what's on the screen when you click a button, and even play games right in your web browser.

It's a language that web developers use to add life and functionality to websites, making them more than just static pages of text and pictures. JavaScript is what makes the web dynamic and fun!

ADVANTAGE

Interactive Websites: JavaScript makes websites come to life. It allows you to add interactive features like buttons, forms, and pop-up messages, so users can do more than just look at a page.

User-Friendly: It helps create a better user experience. For example, you can check if someone entered their email correctly in a form, reducing errors and frustration.

Wide Compatibility: JavaScript works in almost all web browsers. So, no matter if you're using Chrome, Firefox, or another browser, JavaScript will run.

Versatile: You can use JavaScript not only for websites but also for building web applications, mobile apps, and even server-side operations. It's like a multi-tool for web development.

Large Community: Many developers use JavaScript, so there are lots of resources and help available online. If you run into a problem, chances are someone has a solution or advice to offer. It's like having a big group of friends to help you when you need it.

DISADVANTAGE

Security Risks: JavaScript can be used by bad actors to break into websites or steal information, like having an unlocked door that anyone can enter.

Browser Differences: JavaScript may act differently in various web browsers, causing problems for developers, much like a toy that works differently depending on the room you're in.

Slower Loading: Using lots of JavaScript can make web pages load and run slowly, just like a car that becomes sluggish when you put too many heavy things in it.

Needs Internet: JavaScript relies on the internet. If you lose your connection, some features might stop working, similar to a TV that only functions when it's connected to electricity.

Accessibility Challenges: Not everyone can use JavaScript-based features, which can make websites harder to use, similar to having parts of a book written in a language not everyone can understand.

2.2 Back-end Technologies

2.2.1 MongoDB

MongoDB, in simple words, is like a digital filing cabinet for storing information in a flexible and easy-to-access way. Instead of using rows and columns like a traditional spreadsheet, MongoDB uses collections of documents. Each document is like a digital file that can hold various types of data.

Imagine you have a collection of books. In MongoDB, each book would be a document, and you can store information like the title, author, genre, and even comments about the book all in one place. It's a great way to organize and retrieve information for websites, apps, and other data-driven projects. MongoDB is known for its flexibility, scalability, and speed, making it a popular choice for many modern applications.

ADVANTAGES

Flexible Structure: MongoDB doesn't force you to use a fixed structure for your data, unlike traditional databases. You can change how your data looks as your app evolves, making it flexible and adaptable.

Handles Growth: MongoDB can grow with your app. When your app becomes super popular, you can spread your data across many computers to handle all the visitors without slowing down.

Quick Searches: MongoDB can find information super fast, even when you have lots of data. It's like finding a single grain of sand on a huge beach; it doesn't take long.

Familiar Format: MongoDB stores data in a way that's easy for programmers to understand, like a digital version of a paper document. This makes it friendly for developers who work with modern programming languages.

Helpful Community: Lots of people use MongoDB, so there are many friends out there who can help if you have questions or need to do something special with your data. It's like having a big group of friends who know all about organizing things.

DISADVANTAGE

Lacks Transactions: MongoDB doesn't support multi-document transactions in all cases, which can be a problem if you need complex operations that involve multiple pieces of data. It's like trying to do two things at once, but you can only finish one.

Memory Usage: MongoDB can use a lot of computer memory, especially when handling large amounts of data. This could slow down other applications running on the same computer, like having too many apps open on your phone making it sluggish.

Not Suitable for Complex Queries: While MongoDB is great for simple queries, it may struggle with complex ones. It's like asking a simple calculator to solve a really complicated math problem; it might not handle it well.

Data Size: MongoDB's flexibility can sometimes lead to larger data sizes compared to structured databases. Storing extra data can be like carrying around things you don't need in your bag, making it heavier.

2.2.2 ASP.NET

ASP.NET, in simple words, is a framework for building powerful and interactive websites and web applications. It's like a toolbox full of tools that help developers create web pages that can do a lot of different things.

ASP.NET allows you to add features like user registration, online shopping, or social networking to websites. It's like adding special functions to a regular website to make it more useful and fun for users. Developers use languages like C# or VB.NET with ASP.NET to create these web applications. It's like the foundation that helps you build amazing and dynamic websites, making the internet a more exciting place to explore.

ADVANTAGES

Powerful Web Apps: ASP.NET helps developers create powerful web applications with features like user accounts, databases, and interactive forms. It's like giving your website superpowers to do complex tasks.

Security: ASP.NET includes built-in security features to protect your website from hackers and unauthorized access. It's like having a security guard for your online business.

Scalability: It's easy to make your website grow as your business does. You can add more features or handle more visitors without much trouble, like expanding your shop when more customers show up.

Speed: ASP.NET is designed to be fast, so your website loads quickly and keeps users engaged. It's like serving customers at a fast-food restaurant – they don't have to wait long.

Community Support: Many developers use ASP.NET, so there are lots of resources and help available online. It's like having a big group of friends who know how to build great websites and are willing to help you out.

DISADVANTAGES

Complexity: ASP.NET can be complex for beginners. It's like trying to drive a big, powerful car when you're just starting to learn how to drive.

Cost: Some versions of ASP.NET, like ASP.NET Core, are open source and free, but others, like ASP.NET for Windows, can be costly. It's like paying extra for a deluxe version of a product.

Windows Dependency: ASP.NET is closely tied to the Windows operating system, which means it might not work as well on non-Windows servers. It's like having a tool that only fits one type of screw.

Learning Curve: Learning to use ASP.NET effectively can take time, especially if you're new to it. It's like learning a new language; it takes practice.

Limited Hosting Options: Not all web hosting providers support ASP.NET, which can limit your options when choosing where to host your website. It's like wanting to visit a restaurant, but it's only open on certain days.

2.2.3 Python

Python, in simple words, is a computer language that people use to tell computers what to do. It's like a set of instructions you give to your computer, and it understands and follows them.

Python is known for being easy to read and write. It's like using plain English to communicate with your computer. People use Python to build websites, write programs, analyze data, and even make games. It's a versatile and friendly language that makes computers do all sorts of tasks for us.

ADVANTAGES

Easy to Learn: Python has a straightforward and readable syntax, making it easy for beginners to pick up and understand.

Versatile: Python can be used for a wide range of tasks, from web development to data analysis and artificial intelligence.

Large Community: There's a big community of Python users, so you can find plenty of help, tutorials, and resources online.

Ready-Made Tools: Python comes with a lot of built-in functions and libraries, so you don't have to reinvent the wheel when coding.

Cross-Platform: Python works on different operating systems, so you can use it on Windows, macOS, or Linux without major changes.

DISADVANTAGES

Slower Execution: Python can be slower than some other languages because it's an interpreted language, meaning the code is executed line by line rather than compiled into machine code.

Global Interpreter Lock (GIL): Python has a GIL that can limit multi-core CPU utilization, which might affect performance in certain multi-threaded applications.

Not Ideal for Mobile Apps: Python isn't the best choice for mobile app development, as it's not as fast or memory-efficient as languages like Java or Swift.

Less Suitable for High-Performance Computing: For very compute-intensive tasks, Python may not be the best choice, as it's not as fast as languages like C or C++.

Limited in Some Areas: While Python is versatile, it may not be the best choice for specialized tasks like real-time systems or low-level programming, where more performance control is needed.

2.2.4 FireBase

Firebase, in simple words, is like a magical toolbox for building web and mobile apps. It provides developers with a bunch of pre-made tools and services that make it easy to create apps with features like user authentication, real-time updates, storage for files, and databases to store information.

Imagine you want to build a chat app. With Firebase, you don't have to start from scratch. You can use Firebase's tools to handle things like letting users sign in, sending messages in real-time, and storing those messages safely.

It's like having a handy set of building blocks that save you time and effort when creating cool apps, making the app development process smoother and more enjoyable.

ADVANTAGES

Easy to Use: Firebase is designed to be beginner-friendly, making it easy for developers of all levels to build and manage web and mobile apps without extensive backend expertise.

Real-time Updates: Firebase offers real-time database and messaging features, allowing you to create apps that instantly update when new data is available. It's like having a live chat that updates as people type.

Authentication: Firebase provides built-in authentication services, making it simple to add user registration and login functionality to your app. It's like having a secure entryway to your app.

Scalable: Firebase can handle apps with a small number of users or those with millions of users. It scales effortlessly as your app grows, like a room that can expand to accommodate more guests.

Hosting and Cloud Services: Firebase offers hosting for web apps and cloud storage for files, simplifying the deployment and storage process. It's like having a reliable and secure cloud-based storage unit for your app's data and files.

DISADVANTAGES

Limited Control: Firebase abstracts many backend details, which can be limiting if you need fine-grained control over your server infrastructure. It's like having a car with an automatic transmission when you want to shift gears manually.

Scalability Costs: While Firebase is great for small to medium-sized apps, it can become expensive as your app scales up, particularly for high-traffic or data-intensive applications.

Vendor Lock-In: Once you've built your app on Firebase, it can be challenging to migrate to a different platform, leading to vendor lock-in. It's like being in a long-term contract with a specific phone carrier.

Complex Queries: Firebase's database is NoSQL, which can make complex queries challenging compared to traditional relational databases. It's like trying to find specific items in a messy room.

2.3 Technologies used in the project

The technology used in the front end for the project is HTML, CSS due to HTML is like the blueprint of the website. It helps us organize and arrange the content on the site. It's where we put text, images, headings, and links in a way that makes sense. CSS is like the decorator of the website. It helps us choose colors, fonts, and layouts to make the site look attractive and easy to use. It's how we give the website its unique appearance. So, HTML sets up the content, and CSS makes it look good. Together, they create a well-structured and visually appealing home renovation website that visitors can easily understand and enjoy. The technology used in the back end for the project is MongoDB and Firebase due to MongoDB acts like a digital storage room where we keep all the information needed for the website. This includes details about renovation projects, customer reviews, and more. It's flexible and can hold various types of data, just like a well-organized storage space. Firebase is like a communication hub for the website. It helps us manage user accounts, handle real-time messaging between homeowners and renovation teams, and make sure everyone's information is always up-to-date. It's like a digital command center that keeps everything running smoothly.

Chapter 3 - Requirement and Analysis

3.1 Problem Definition

Home Revive is a game-changer for homeowners looking to renovate. It simplifies and enhances the entire renovation process by consolidating all the essential elements into one user-friendly platform.

The platform boasts a vast catalog of innovative designs that cater to various styles and tastes. This extensive library makes it effortless for homeowners to discover creative inspiration. Whether you're dreaming of a modern, sleek look or a cozy, traditional ambiance, Home Revive has you covered.

What truly sets Home Revive apart is its comprehensive approach. In addition to providing designs and customer reviews, the platform offers a carefully curated collection of furniture. This ensures that every aspect of your home aligns seamlessly with your vision, creating a cohesive and harmonious living space.

The result? Home Revive saves you invaluable time, money, and stress. It empowers you to embark on your home renovation journey with confidence and ease, turning what was once a daunting task into an enjoyable adventure. It's the ultimate destination where ideas, professionals, and furnishings converge to bring your dream home to life.

With Home Revive, you can bid farewell to renovation headaches and welcome the future of home improvement. Your dream home is just a click away, waiting for you to explore and make a reality.

3.2 Requirement Analysis

Home Revive fulfills a variety of requirements for homeowners and individuals embarking on the home renovation journey. Here are the key requirements it addresses:

- **Design Inspiration**: Home Revive offers an extensive catalog of innovative designs, catering to diverse styles and tastes. It provides homeowners with a wealth of ideas and inspiration for their renovation projects.
- **Professional Services**: The platform connects users with reputable architects, designers, and contractors, simplifying the process of finding and hiring skilled professionals for their renovation projects.

- **Customer Reviews**: Home Revive allows users to access and review customer feedback and ratings for architects, contractors, and other professionals, helping homeowners make informed decisions.
- **Visual Aids:** The website offers a wealth of photos and visual aids, allowing users to visualize their renovation ideas and see real-world examples of completed projects.
- **Furniture Selection**: Home Revive provides a curated collection of furniture options, helping homeowners find the perfect furnishings to complement their renovation designs.
- **Time and Cost Savings**: By consolidating all these essential elements in one place, Home Revive saves homeowners time and effort that would otherwise be spent searching for various services and products.
- **Ease of Use**: The user-friendly platform simplifies the renovation process, making it accessible to a wide range of homeowners, from novices to experienced renovators.
- **Comprehensive Approach:** Home Revive takes a holistic approach to home renovation, ensuring that every aspect of the project, from design to execution to furnishing, aligns seamlessly with the homeowner's vision.
- **Confidence and Empowerment**: By providing a one-stop solution, Home Revive empowers homeowners to confidently embark on their renovation projects, reducing the stress and uncertainty typically associated with home improvement.

In summary, Home Revive fulfills the requirements of simplifying, enhancing, and streamlining the entire home renovation process, making it a valuable resource for anyone looking to transform their living space.

3.3 Planning and Scheduling

3.3.1 Planning

"Home Revive" is a one-stop home renovation website simplifying the renovation process. Here's how it works:

User Registration: Users start by creating an account, allowing them to access the full range of services.

Select Renovation Category: Customers can choose the area of their home they wish to renovate - whether it's the kitchen, bedroom, living room, bathroom, or any other space.

Design Inspiration: Within their selected category, users can browse a variety of design options. They can explore different styles, layouts, and themes to find the one that suits their taste.

Architect Profiles: "Home Revive" features a list of experienced architects renowned for their work. Users can view portfolios and photo galleries showcasing previous projects, including before-and-after pictures.

Budget Flexibility: Customers can set their budget preferences, helping them find designs and architects that align with their financial constraints.

FAQ and Reviews: A detailed FAQ section addresses common queries, ensuring users have all the information they need. Users can also access reviews and testimonials to gauge the quality of service.

Furniture Selection: The website offers a dedicated page for furniture design. Users can browse through a wide range of modern and vintage furniture options tailored to their preferences. Conveniently, they can make purchases directly from the website.

Booking and Payment: Once customers have made their choices, they can proceed to book the services of their chosen architect and finalize the furniture selections. The website provides a secure payment gateway for hassle-free transactions.

"Home Revive" streamlines the entire renovation journey, empowering users to make informed decisions, access expert guidance, and conveniently purchase furniture items, all within a single platform. This simplification of the renovation process saves time, eliminates guesswork, and ensures that customers can bring their dream home to life with ease and confidence.

3.3.2 Scheduling

3.3.2.1 Gantt chart

A Gantt chart is like a visual to-do list for projects. It helps people see what needs to be done, when it should happen, and in what order. Here's how it works:

List of Tasks: Imagine you're planning a party. You'd make a list of all the things to do, like buy decorations, send invitations, and bake a cake. Each task goes on your list.

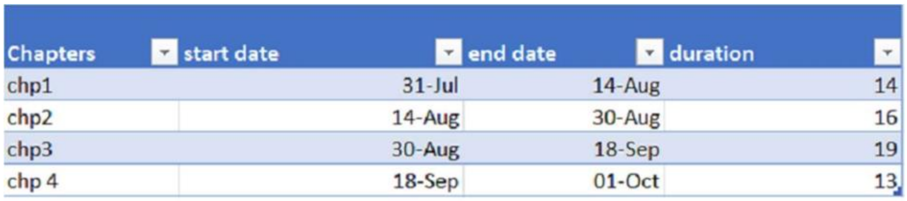
Timeline: Next, you decide when each task should happen. For instance, you might buy decorations two weeks before the party and bake the cake the day before.

Bars on a Timeline: Now, picture a calendar with a timeline. You draw bars or blocks on this timeline to represent each task. The length of each bar shows how long a task takes, and its position on the timeline tells you when it starts and ends.

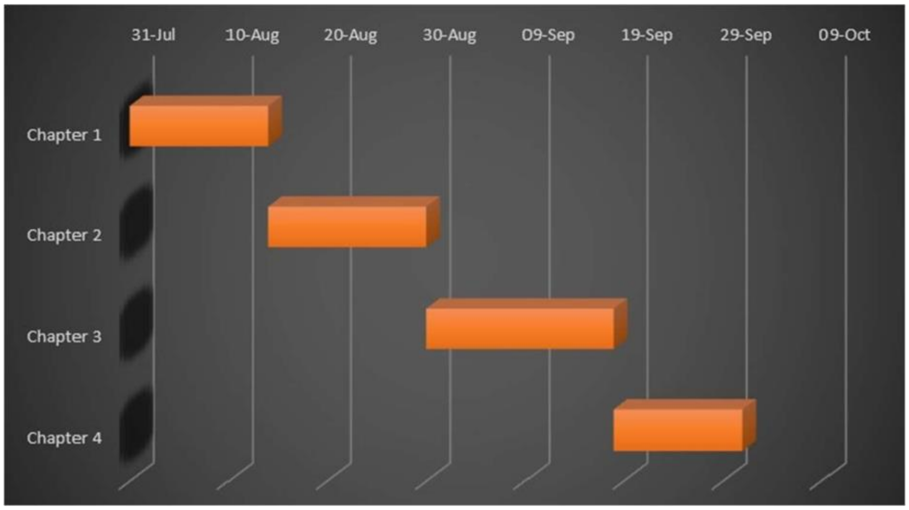
Dependencies: Some tasks depend on others. For example, you can't send invitations until you've bought them. Gantt charts use arrows to show these connections.

Tracking Progress: As you work on your party, you can color in the bars to show what's done and what's left to do. This way, you always know if you're on schedule.

In simple terms, a Gantt chart helps you plan and keep track of your tasks and when they need to happen, making it easier to manage projects, whether it's a party or something more complex.



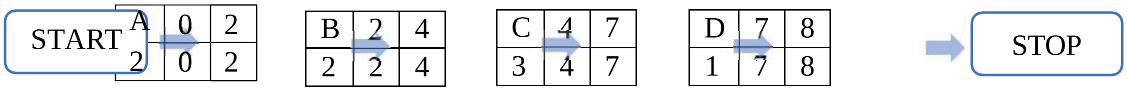
Chapters	start date	end date	duration
chp1	31-Jul	14-Aug	14
chp2	14-Aug	30-Aug	16
chp3	30-Aug	18-Sep	19
chp 4	18-Sep	01-Oct	13



Pert Chart

PERT charts are used to effectively plan and manage complex projects. They offer a visual representation of tasks, their dependencies, and time estimates, making it easier to understand project structure. By identifying the critical path, project managers can prioritize tasks, allocate resources wisely, and mitigate risks, ultimately ensuring that the project is completed on time and within budget. PERT charts also aid in communication, tracking progress, and making data-driven decisions, making them a valuable tool in the world of project management.

Activity	Precedence	Duration (Weeks)
A	**-**	**2 Weeks**
B	**A**	**2 Weeks**
C	**B**	**3 Weeks**
D	**C**	**1 Week**



Slack = 0 for all the activities
Therefore, Critical Path = A – B – C – D

3.4 Hardware and Software Requirement

3.4.1 Hardware Requirement

- **Edition** : Windows 11 Home
- **Processor** : ntel(R) Core(TM) i5-8265U CPU @ 1.60GHz   1.80 GHz
- **System type** : 64-bit operating system, x64-based processor
- **RAM** : 8.00 GB (7.89 GB usable)
- **Storage** : 476 GB
- **Version** : 22H2
- **OS build** : 22621.2134

3.4.2 Software Requirement

- Andriod Studio
- Java Development Kit
- Visual Studio Code
- React
- Mongo DB

3.5 Conceptual Model

3.5.1 ERD Diagram

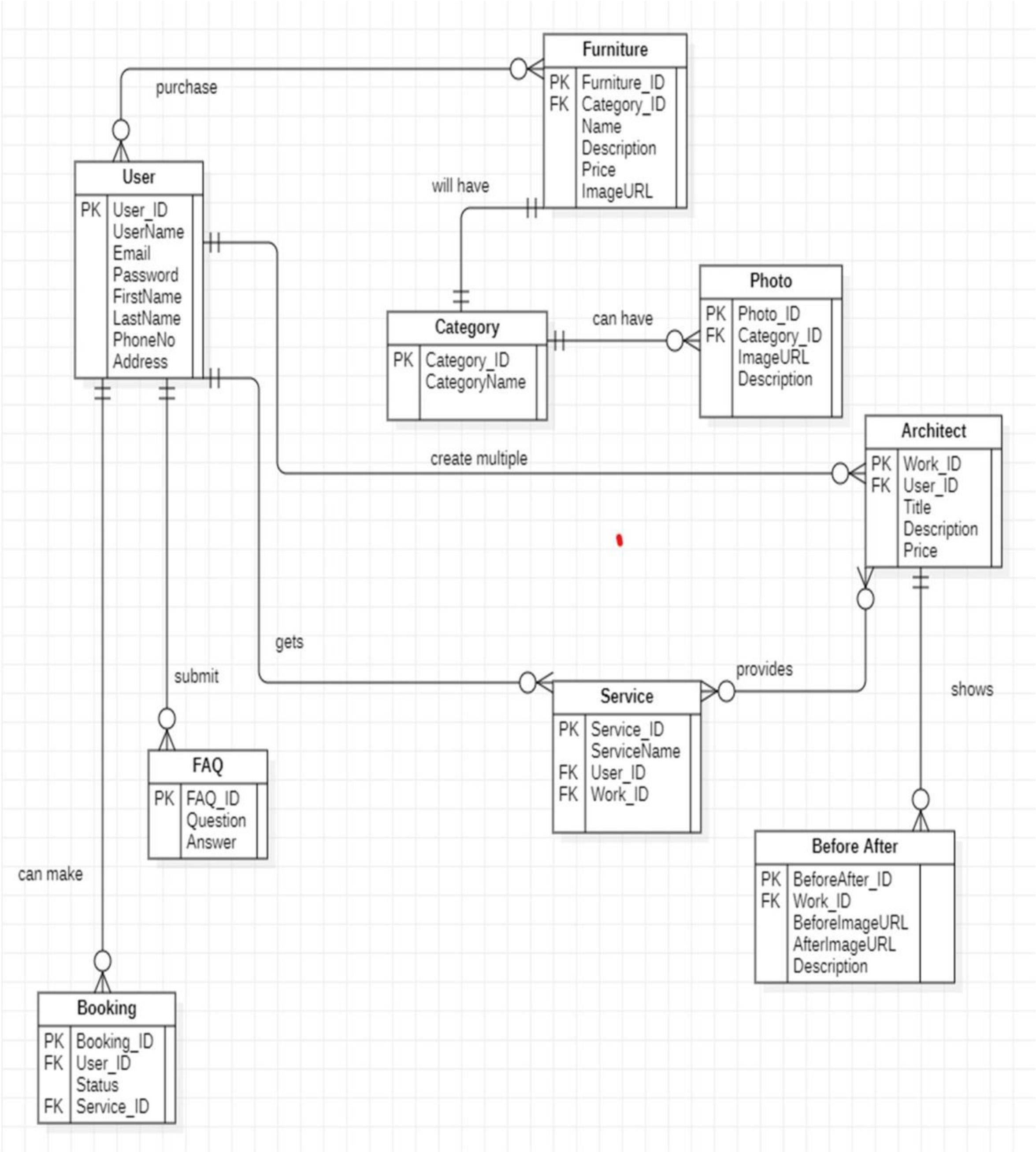


Figure Entity-Relationship Diagram

3.5.2 Date Flow Diagram

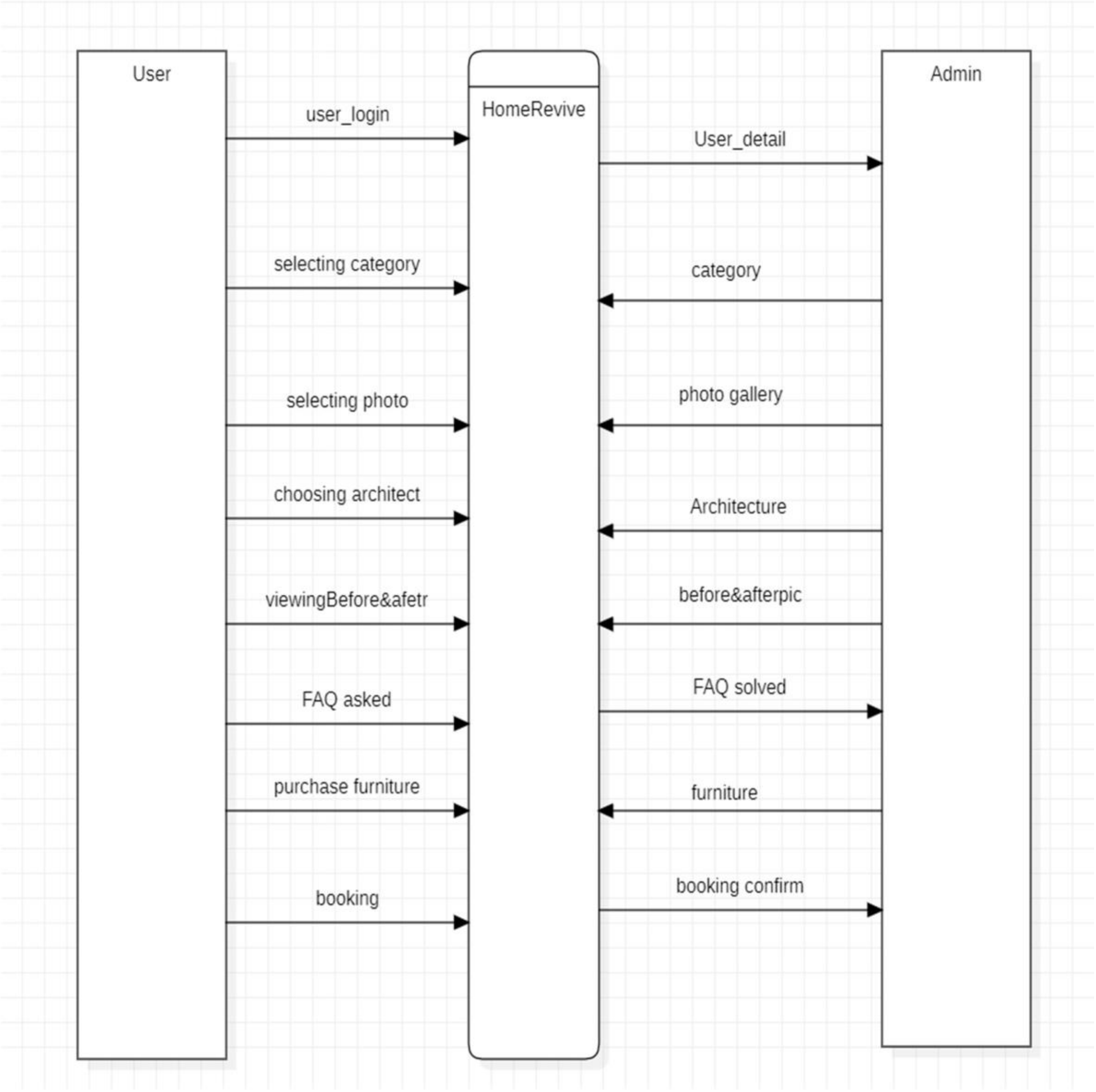


Fig 3.5.2.1 Level 0

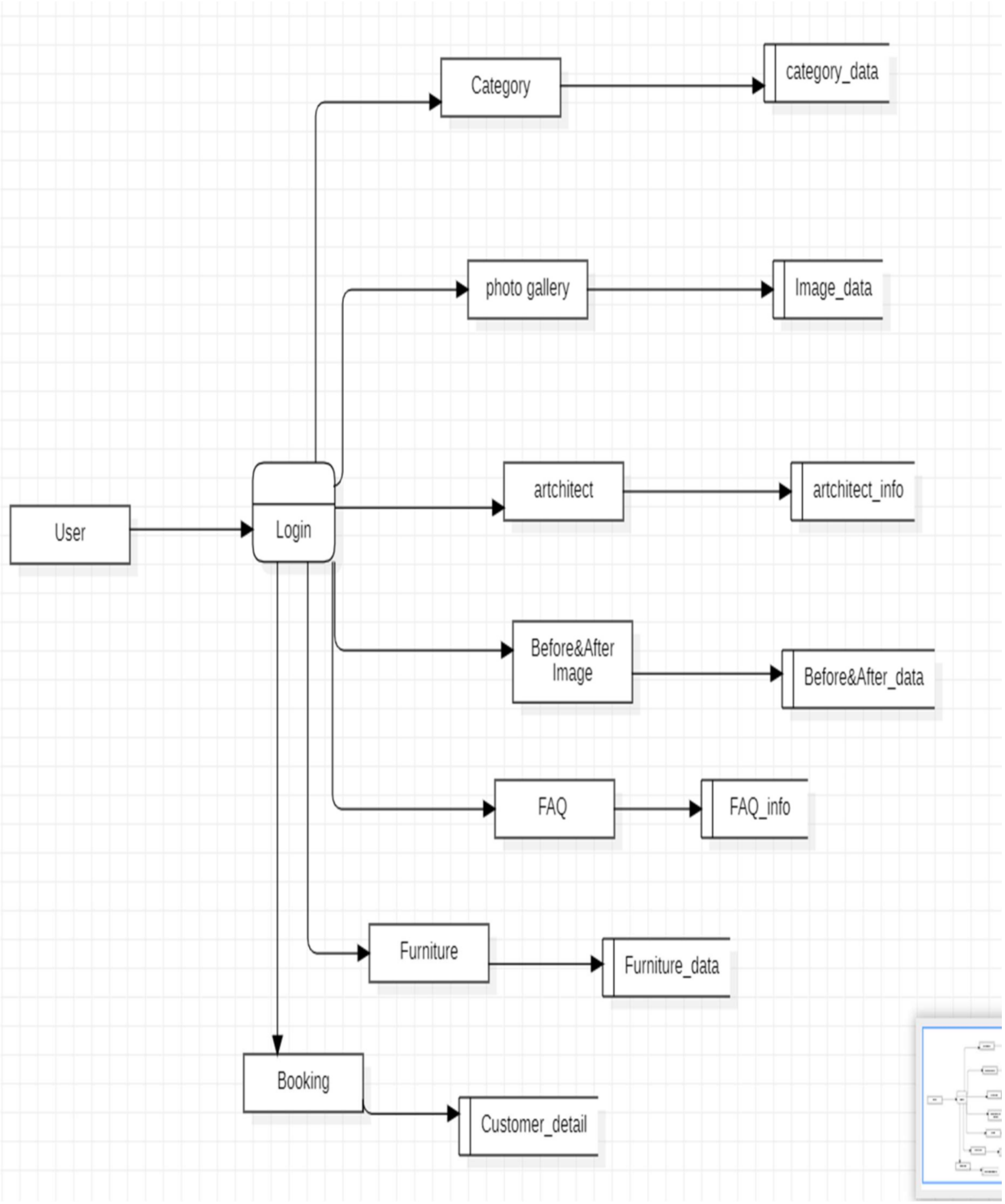


Fig 3.5.2.2 Level 1

Chapter 4: System Design

4.1. Basic Modules

The application will include the following modules-

- **User module** - This module enables individuals to create accounts by providing their personal information, such as name, email, and password. This information is essential for uniquely identifying and associating users with their profiles.
- **Category -** The Category Module in HomeRevive categorizes the options like bedroom, kitchen, living room etc simplifying users' search for space-specific designs in styles like modern, vintage, and contemporary. Visual aids inspire informed decisions, while collaboration with professionals allows users to actualize their ideas for a personalized and stunning home. It's a user-friendly source of design inspiration and implementation.
- **Photos -** The Photos Module within HomeRevive is a visual showcasing the designs photos by which they can select design outlets pictures of completed renovation projects. Users can view these images to gain inspiration and a clear understanding of the transformative power of different design choices. This module not only serves as a source of creative ideas but also helps users make informed decisions about their own home improvement projects by providing real-world examples of successful transformations.
- **Architect -** The Architect Module in HomeRevive facilitates users in collaborating with experienced architects to bring their design visions to life. Users can communicate their ideas, preferences, and requirements to these professionals for personalized design solutions. Architects provide expertise and guidance, ensuring that renovation projects are executed effectively.
- **Before and After -** The Before and After Module in HomeRevive showcases a visual transformation of spaces before and after renovation projects. Users can see the dramatic changes in design, layout, and aesthetics, providing inspiration and a tangible preview of what their own projects could achieve.
- **FAQ -** The FAQ (Frequently Asked Questions) Module in HomeRevive offers a curated list of common user queries and their answers, addressing key

concerns and providing clarity. It serves as a convenient resource for users to find quick and relevant information about the website's services, policies, and processes.

- **Furniture module –** The furniture module provides users with a comprehensive catalog of furniture items, including sofas, tables, chairs, cabinets, and more. These items are curated to match various design styles available on the website, making it easier for users to find furniture that aligns with their design preferences.
- **Booking -** The Booking Module in HomeRevive enables users to schedule renovation services seamlessly. Users can select their preferred dates and specify project details, streamlining the booking process. This module ensures efficient communication between users and service providers.
- **Service -** The Service Module in HomeRevive provides users with information about the range of renovation services available. It includes details about service offerings, such as interior design, construction, remodeling, and more. Users can explore these services to find the right solutions for their home improvement needs.

4.2 Schema Design

4.2.1 User Table

Attributes	Datatype	Relationship
User_ID	Integer(10)	PK
UserName	Varchar(50)	
Email	Varchar(30)	
Password	Varchar(20)	
FirstName	Varchar(50)	
LastName	Varchar(50)	
Address	Varchar(50)	

4.2.2 Category Table

Attribute	Datatype	Relationship
Category_ID	Integer(10)	PK
CategoryName	Varchar(20)	

4.2.3 Photos Table

Attribute	Datatype	Relationship
Photo_ID	Integer(10)	PK
Category_ID	Integer(10)	FK
ImageURL	Varchar(30)	
Description	Varchar(50)	

4.2.4 Architect Table

Attribute	Datatype	Relationship
Work_ID	Integer(10)	PK
User_ID	Integer(10)	FK
Title	Varchar(40)	
Description	Varchar(50)	
Price	Integer(30)	

4.2.5 Before and After Table

Attribute	Datatype	Relationship
BeforeAfter_ID	Integer(10)	PK
Work_ID	Integer(10)	FK
BeforeImageURL	Varchar(40)	
AfterImageURL	Varchar(40)	
Description	Varchar(50)	

4.2.6 Furniture Table

Attribute	Datatype	Relationship
Furniture_ID	Integer(10)	PK
Category_ID	Integer(10)	FK
Name	Varchar(30)	
Description	Varchar(50)	
Price	Integer(30)	
ImageUrl	Varchar(30)	

4.2.7 FAQ Table

Attribute	Datatype	Relationship
FAQ_ID	Integer(10)	PK
Question	Varchar(50)	
Answer	Varchar(50)	

4.2.8 Booking Table

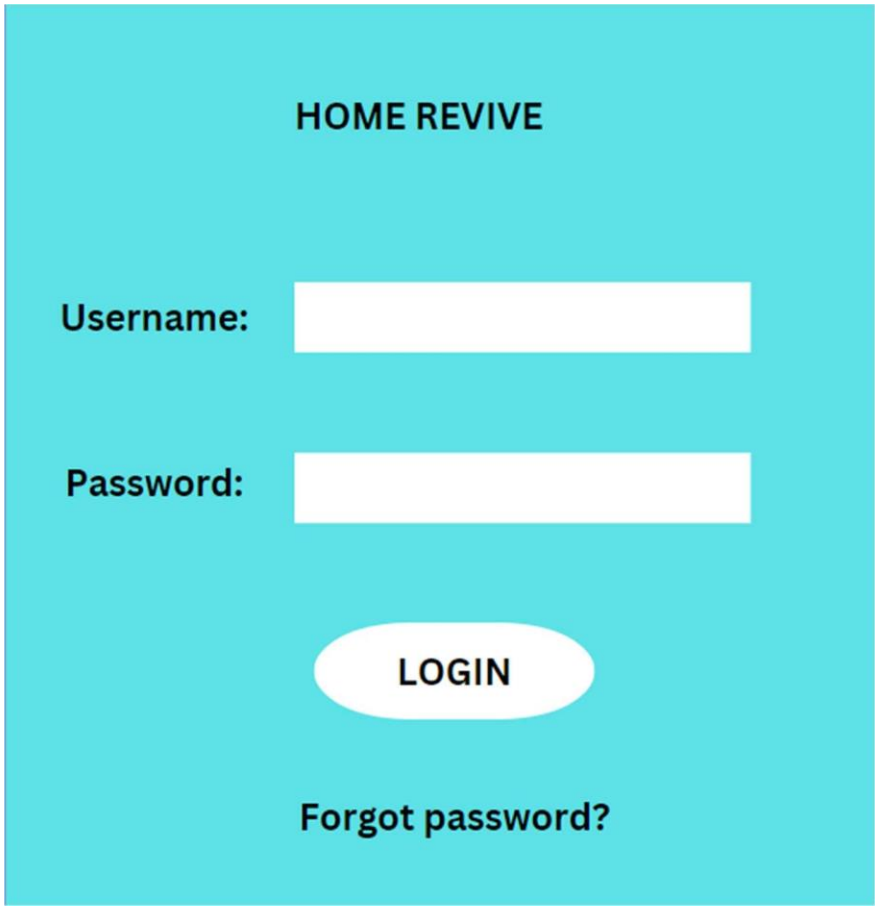
Attribute	Datatype	Relationship
Booking_ID	Integer(10)	PK
User_ID	Integer(10)	FK
Status	Varchar(30)	
Service_ID	Integer(10)	FK

4.2.9 Service Table

Attribute	Datatype	Relationship
Service_ID	Integer(10)	PK
ServiceName	Varchar(30)	
User_ID	Integer(10)	FK
Work_ID	Integer(10)	FK

4.3 User Interface Design

4.3.1. Login

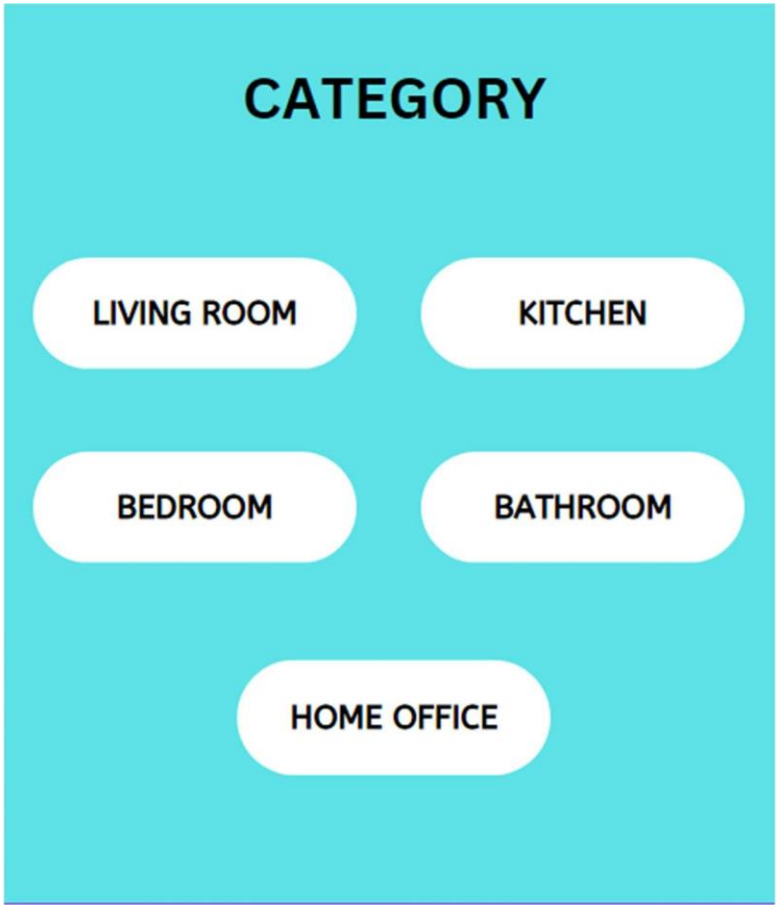


HOME REVIVE

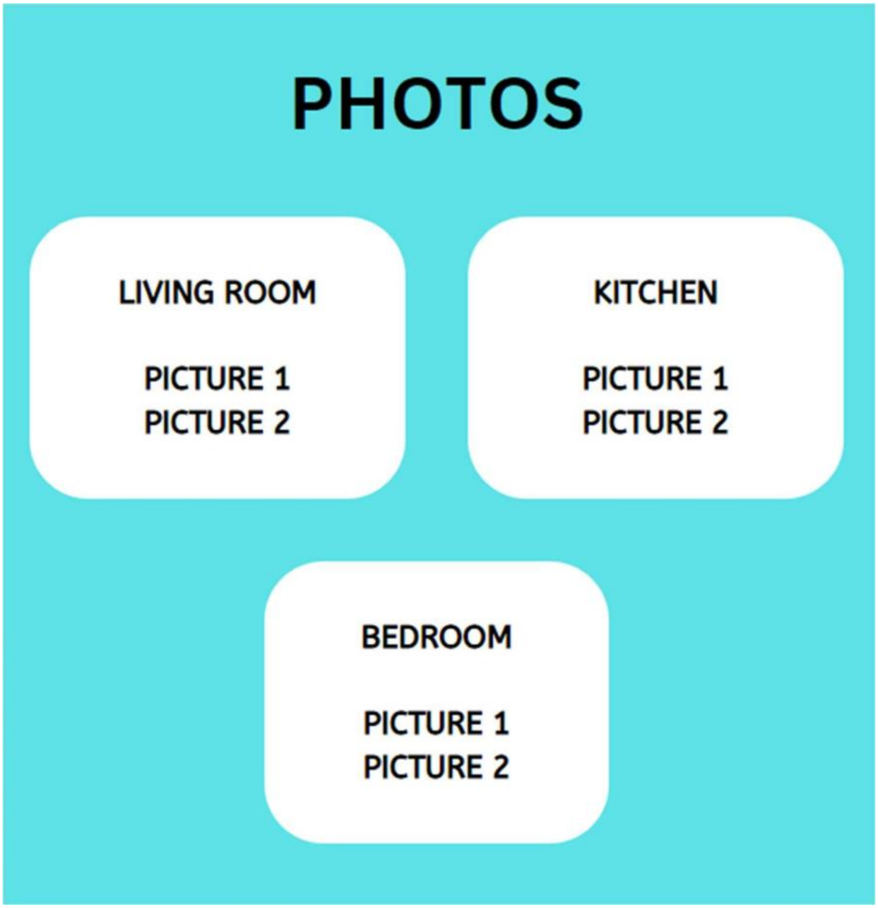
Username:

Password:

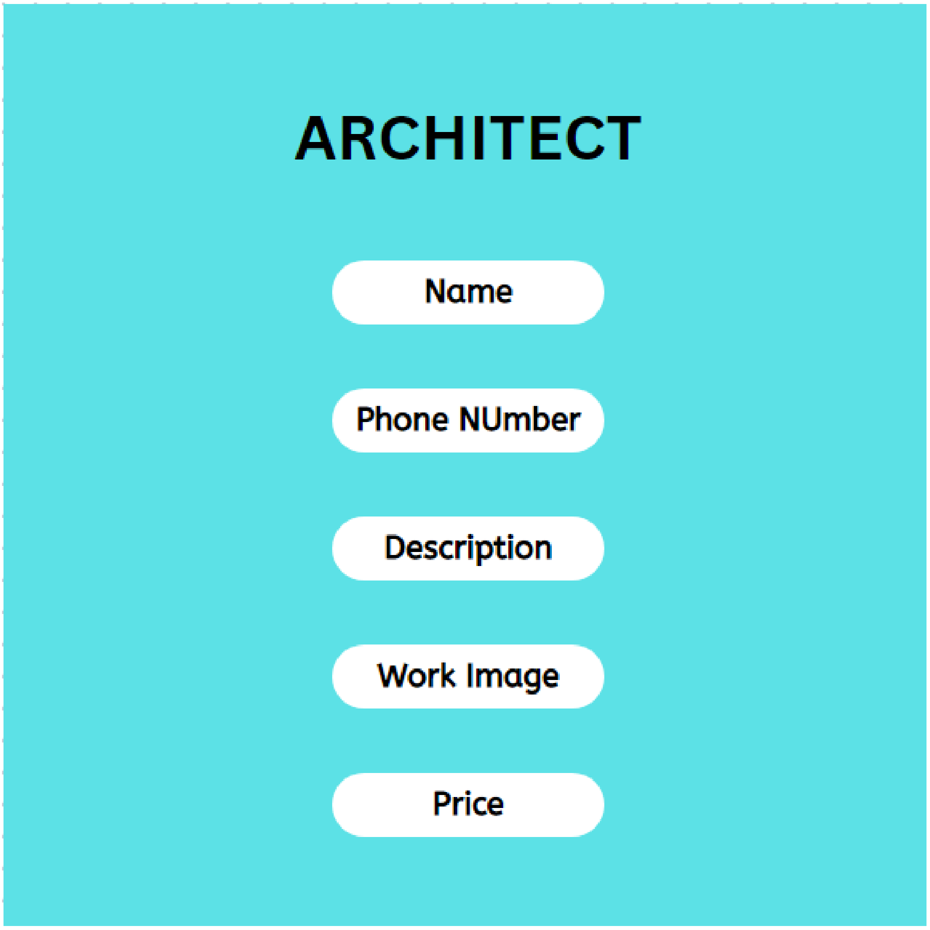
4.3.2. Category



4.3.3. Photos


PHOTOS
LIVING ROOM
PICTURE 1
PICTURE 2
KITCHEN
PICTURE 1
PICTURE 2
BEDROOM
PICTURE 1
PICTURE 2

4.3.4. Architect



ARCHITECT

Name

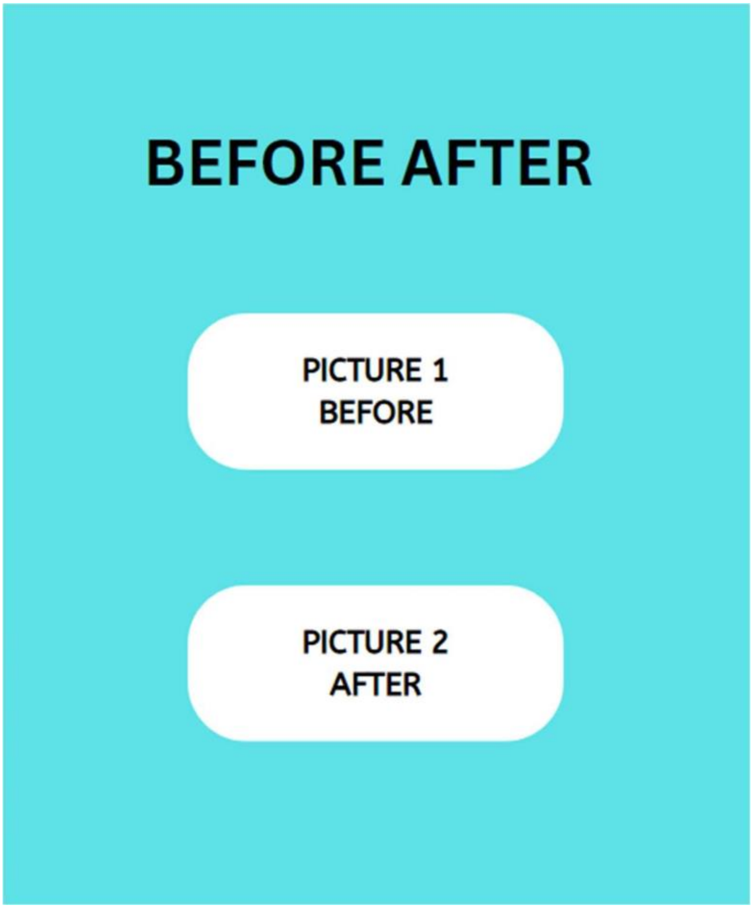
Phone NUmber

Description

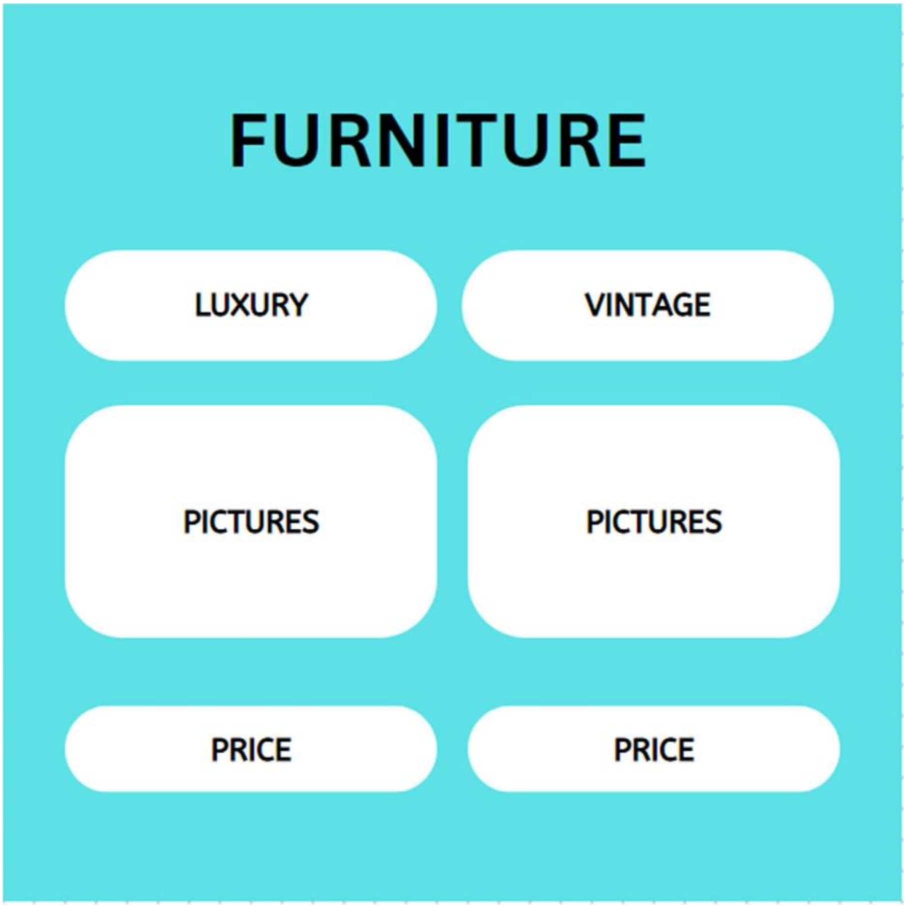
Work Image

Price

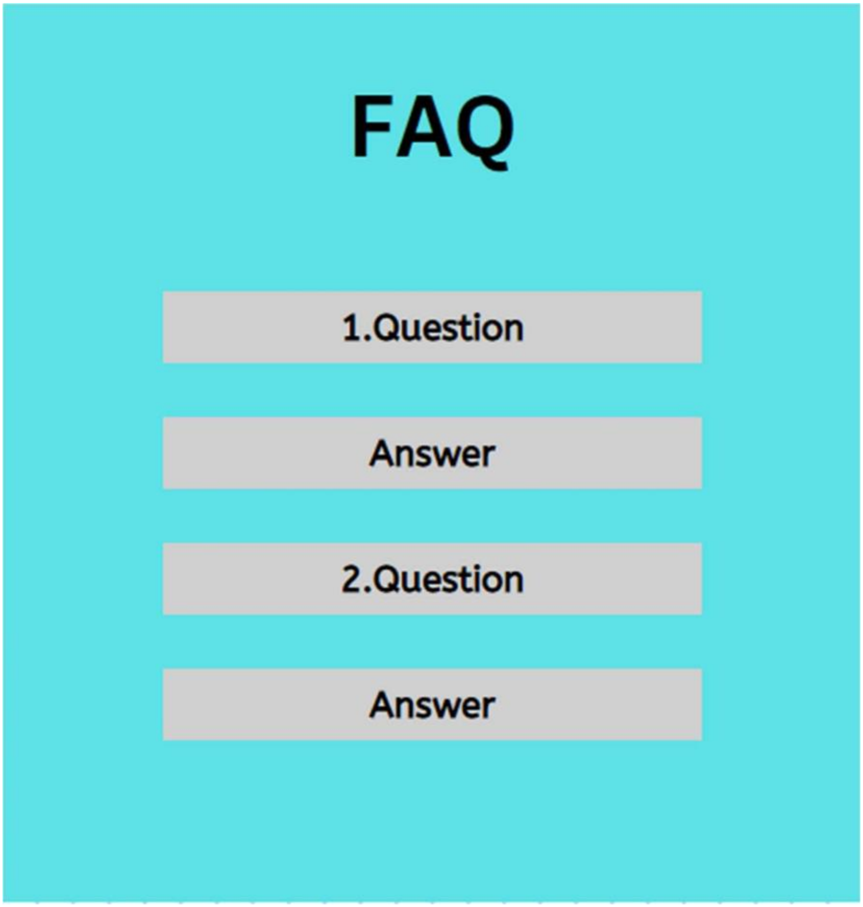
4.3.5. Before and After



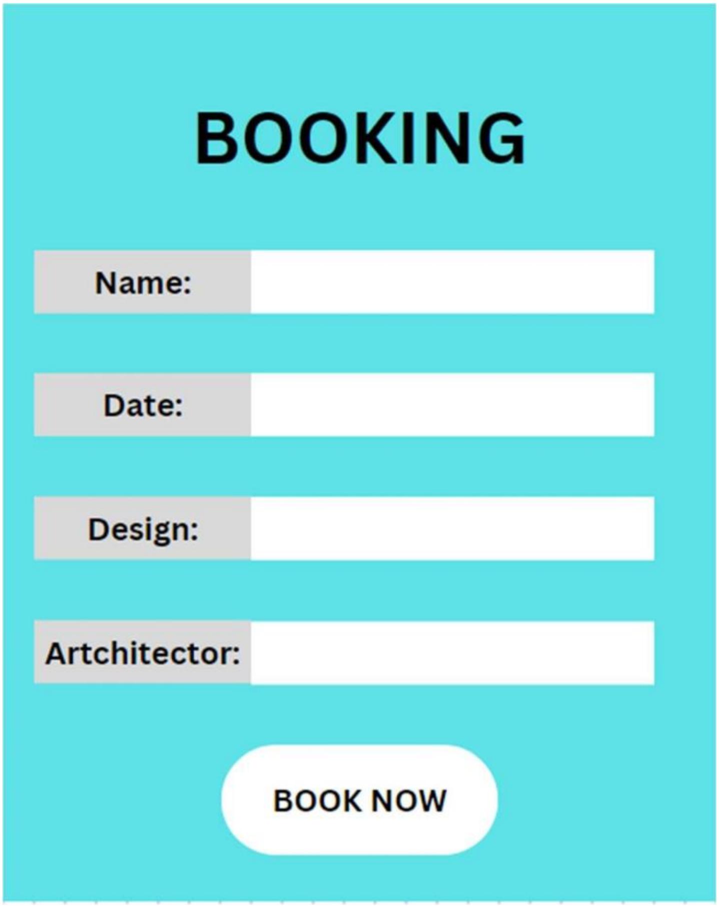
4.3.6. Furniture



4.3.7. FAQ



4.3.8. Booking



4.4 Security Issues

HomeRevive places a strong emphasis on safeguarding your sensitive information. They employ robust security measures to shield your personal data, such as passwords and payment details, from unauthorized access. This includes encrypting your information during transmission and secure storage to ensure it remains confidential.

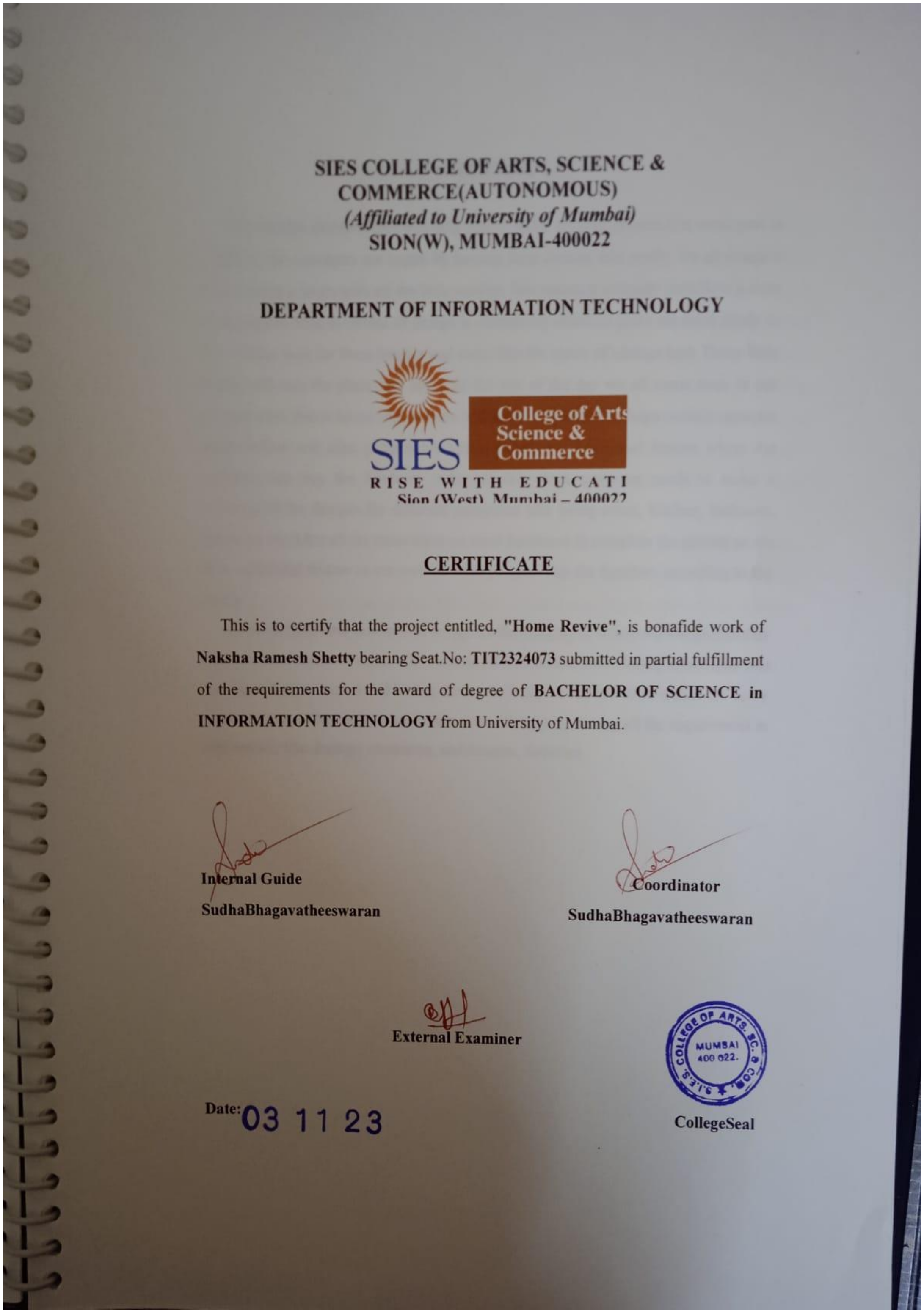
Furthermore, HomeRevive implements stringent user authentication procedures, verifying your identity to prevent unauthorized account access. When it comes to payments, they adhere to strict security protocols to protect your financial information. Regular security assessments are conducted to identify and resolve potential vulnerabilities, and they maintain transparent practices about data handling, ensuring your peace of mind while using their website. Your experience on HomeRevive is consistently secure and trustworthy, thanks to these diligent security efforts.

Reference

https://chat.openai.com/

https://www.canva.com/

https://www.google.com/



SIES COLLEGE OF ARTS, SCIENCE &
COMMERCE(AUTONOMOUS)
(Affiliated to University of Mumbai)
SION(W), MUMBAI-400022

DEPARTMENT OF INFORMATION TECHNOLOGY

SIES
College of Arts
Science &
Commerce
RISE WITH EDUCATI
Sion (West) Mumbai – 400022

CERTIFICATE

This is to certify that the project entitled, **"Home Revive"**, is bonafide work of **Naksha Ramesh Shetty** bearing Seat.No: **TIT2324073** submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai.

Internal Guide
SudhaBhagavatheeswaran

Coordinator
SudhaBhagavatheeswaran

External Examiner

Date: 03 11 23

S.I.E.S. COLLEGE OF ARTS, SC. & COM.
MUMBAI 400 022.

CollegeSeal

Table of Contents

Changes Made In Chapters 1, 2, 3 & 4	
3.3.2 Gant Chart	
3.5.1 Entity RelationShip Diagram	
3.5.2Data Flow Diagram	
4.1 Modules	
4.2 System Design	
4.3 User Interface Design	
5. Chapter: Implementation and Testing	63
5.1. Implementation Approaches	63
5.2. Coding Details and Code Efficiency	63
5.2.1 Code	63
5.2.2. COCOMO	94
5.3 Testing Approaches	96
5.4 Test Cases	96
5.4.1 Login Test Cases	96
5.4.2 Register Test Cases	99
5.4.3 Booking Test Cases	101

6. Chapter: Results and Discussion	102
6.1 Test Report	102
6.2 User Documentation	103
6.2.1 Register Module	105
6.2.2 Login Module	106
6.2.3 Category Module	108
6.2.4 Architect Module	110
6.2.5 Photo Module	114
6.2.6 Furniture Module	115
6.2.7 FAQ Module	117
6.2.8 Booking Module	
6.2.9 Thank You Page	
6.2.10 Your Bookings	
6.2.11 Admin Dash Board	
My Bookings Page	117
Profile Settings	118
Logout	118
7. Chapter: Conclusion	124
7.1. Significance	124
7.2 Limitations	125
7.3 Future scope	125

List of Tables

Table 5.1 COMOCO Model	94
Table 5.2 Signup module Test Cases	96
Table 5.3 Login module Test Cases	98
Table 5.4 Booking Test Cases	101

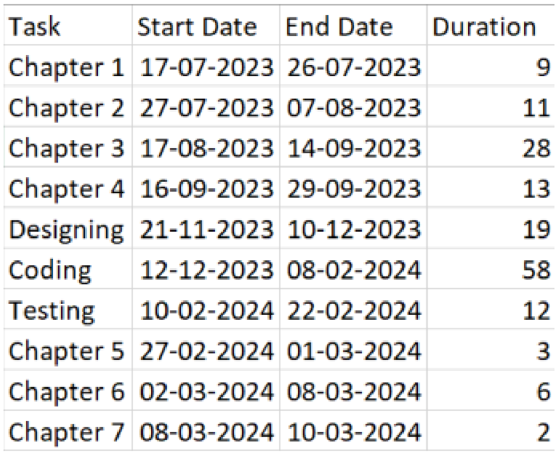
List of Figures

6.2.1 Register Page	104
6.2.2 Login Page	105
6.2.3 Category Page	106
6.2.4 Architect Page	107
6.2.5 Photo Page	110
6.2.6 Furniture Page	111
6.2.7 FAQ Page	112
6.2.8 Booking Page	114
6.2.9 ThankYou Page	115
6.2.10 Your Booking Page	117
6.2.11 Admin Page	117

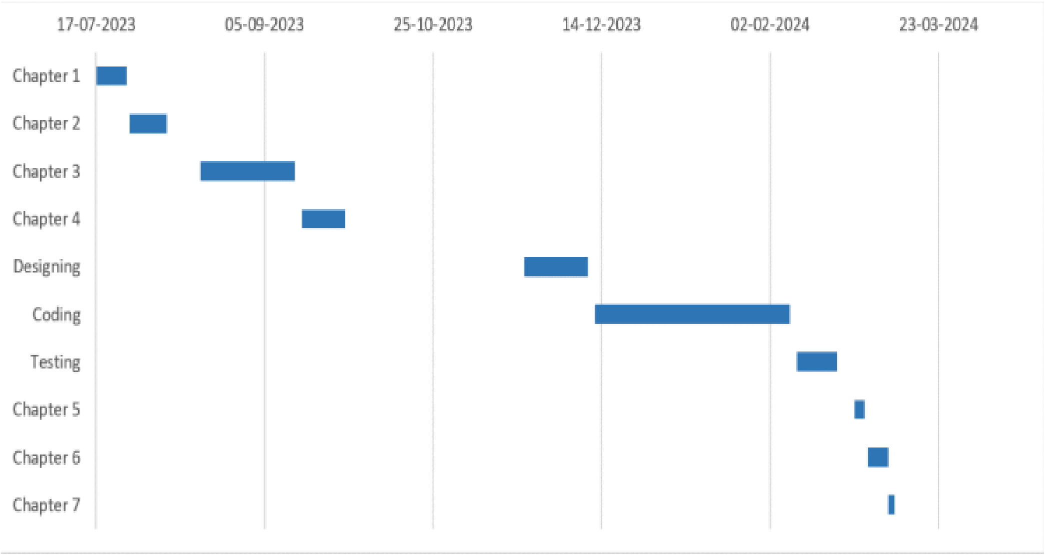
Changes Done in 3,4

Chapter 3: Requirement and Analysis

3.3.2 Scheduling: Gantt Chart:



Task	Start Date	End Date	Duration
Chapter 1	17-07-2023	26-07-2023	9
Chapter 2	27-07-2023	07-08-2023	11
Chapter 3	17-08-2023	14-09-2023	28
Chapter 4	16-09-2023	29-09-2023	13
Designing	21-11-2023	10-12-2023	19
Coding	12-12-2023	08-02-2024	58
Testing	10-02-2024	22-02-2024	12
Chapter 5	27-02-2024	01-03-2024	3
Chapter 6	02-03-2024	08-03-2024	6
Chapter 7	08-03-2024	10-03-2024	2


17-07-2023
05-09-2023
25-10-2023
14-12-2023
02-02-2024
23-03-2024
Chapter 1
Chapter 2
Chapter 3
Chapter 4
Designing
Coding
Testing
Chapter 5
Chapter 6
Chapter 7

3.5 Conceptual Models

3.5.1 Entity Relationship Diagram

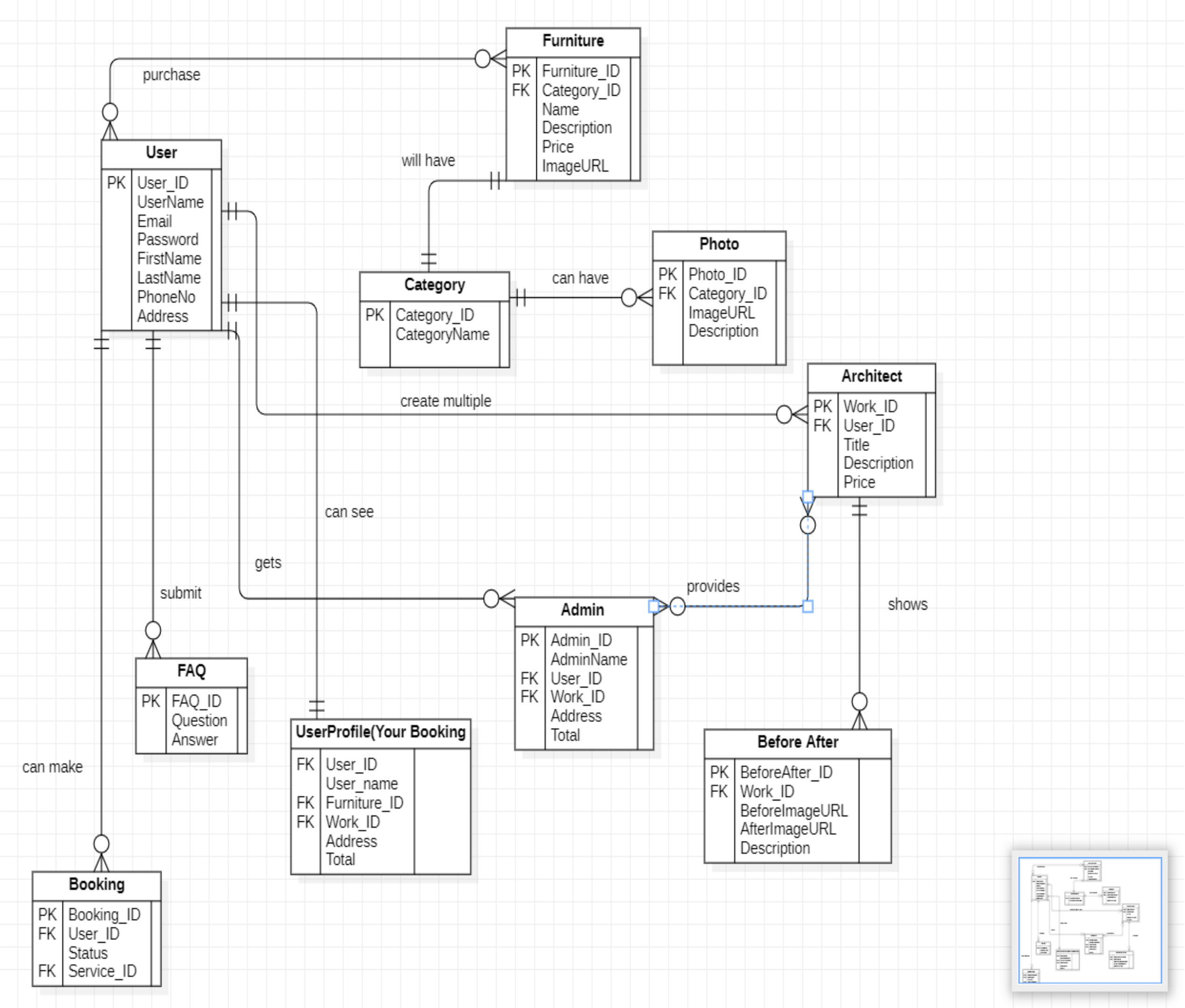


Fig 3.5.1 Entity-Relationship Diagram

3.5.2. Data Flow Diagram

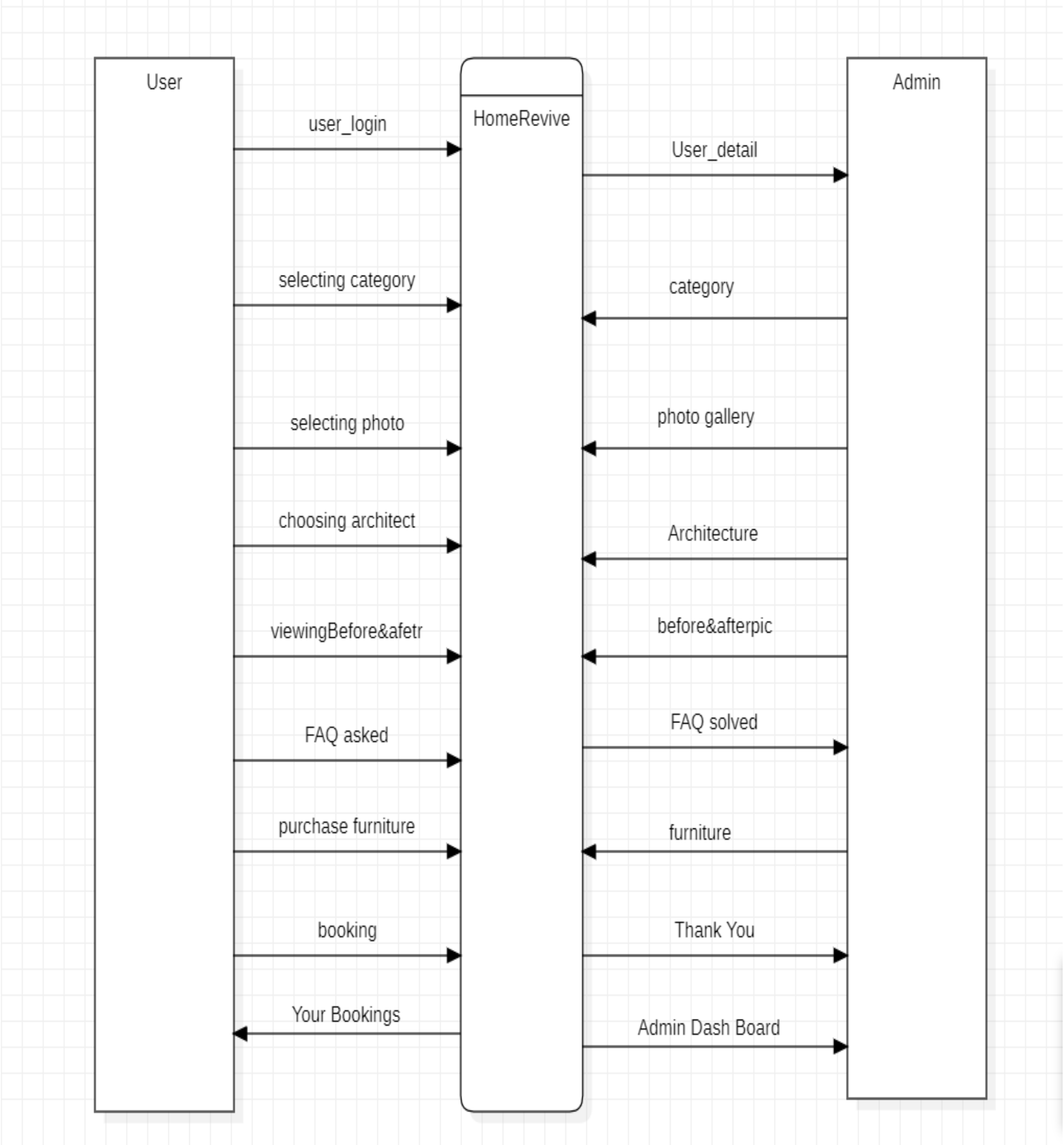
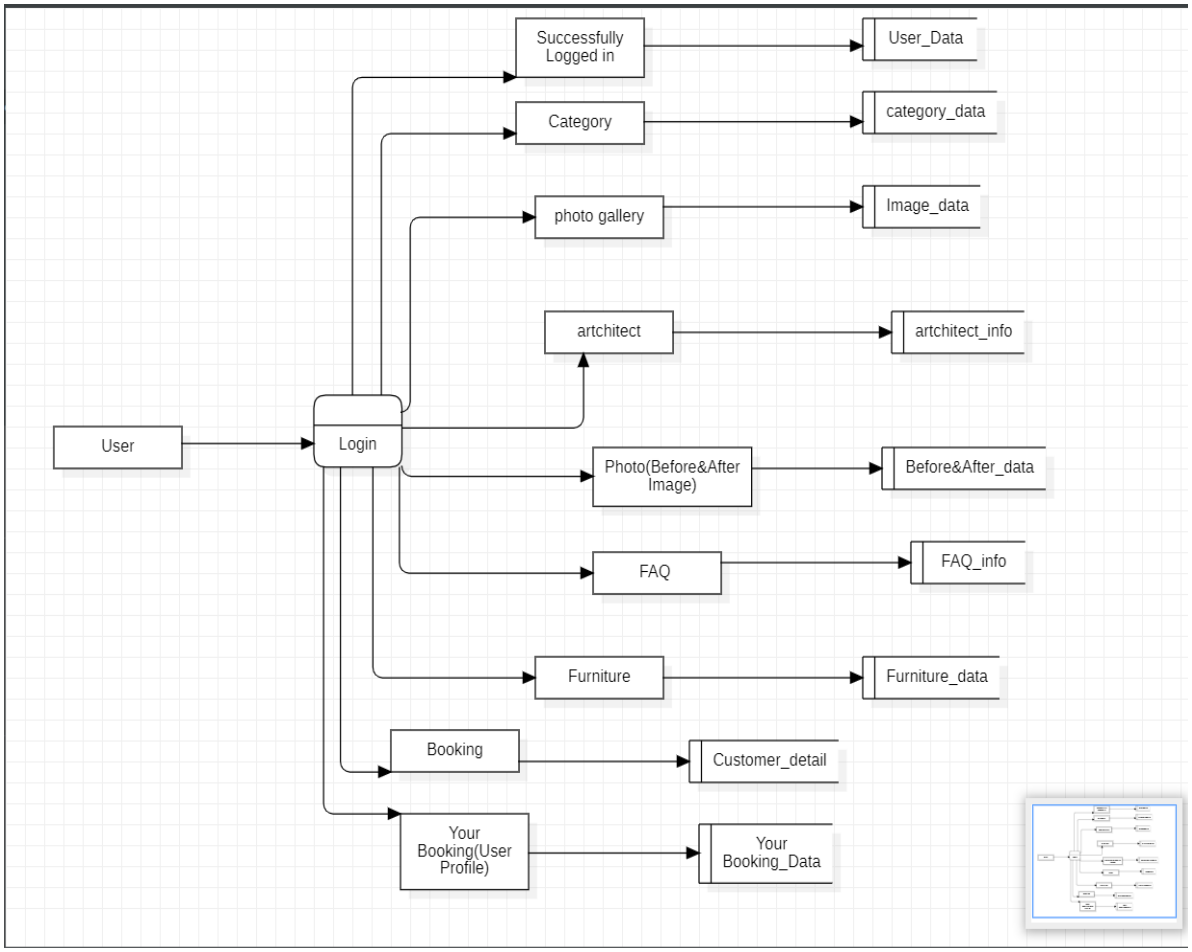


Fig 3.5.2.1 Level 0 Data Flow Diagram



3.5.2.2 Level 1 Data Flow Diagram

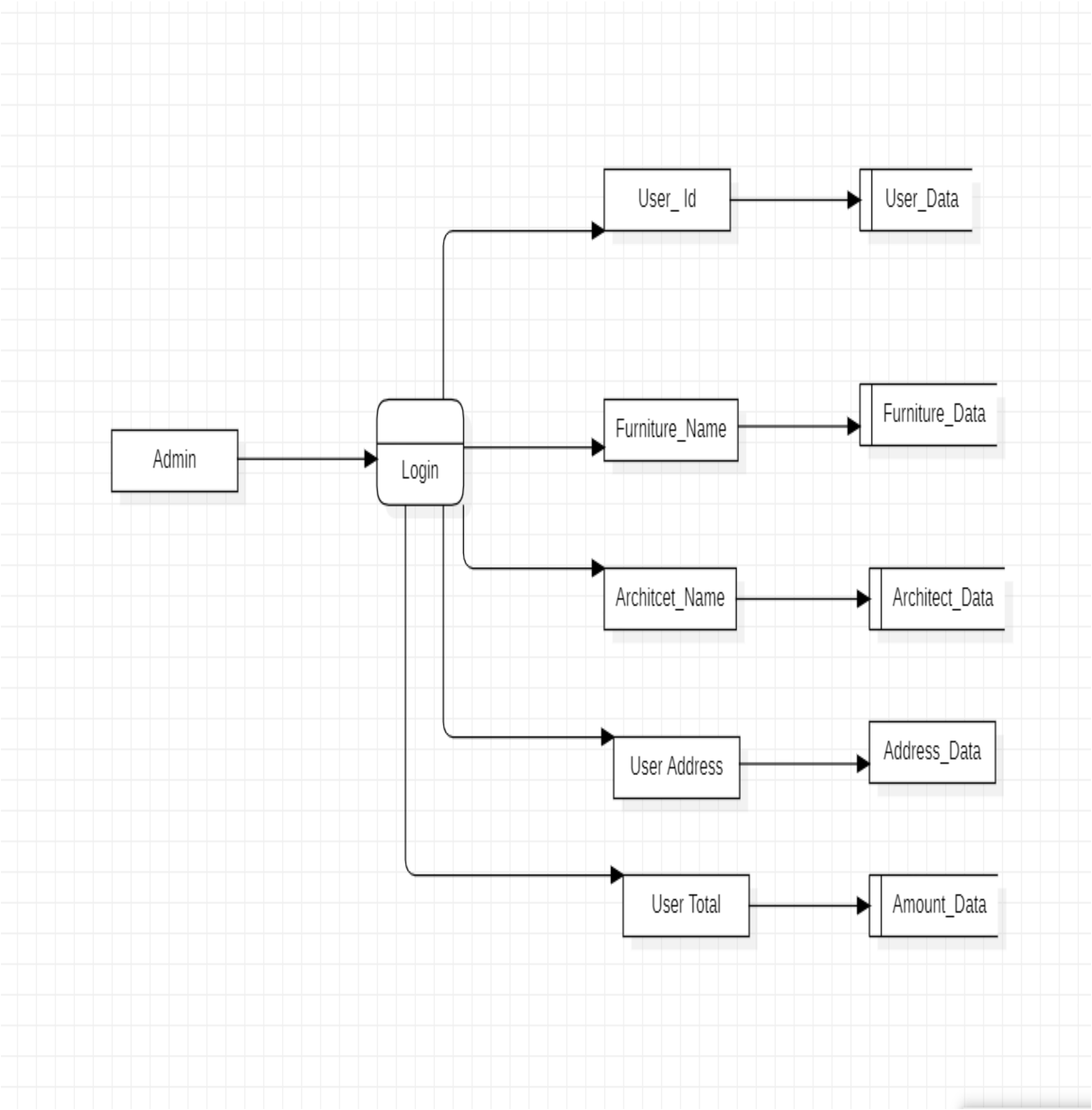


Fig 3.5.2.3 Level 2 Data Flow Diagram

Chapter 4: System Design

4.1 Modules

- **Your Bookings** - HomeRevive's "Your Bookings" section gives users a clear summary of their past and upcoming renovation appointments, making it easy to stay organized and manage their projects effectively. With all booking details in one place, users can ensure a smooth and hassle-free home improvement journey.
- **AdminDashBoard -** The Admin Dashboard page provides administrators with a simple overview of all user bookings, offering easy access to essential information for efficient management and tracking of reservations within HomeRevive.

4.2 System Design:

4.2.10 Your Booking

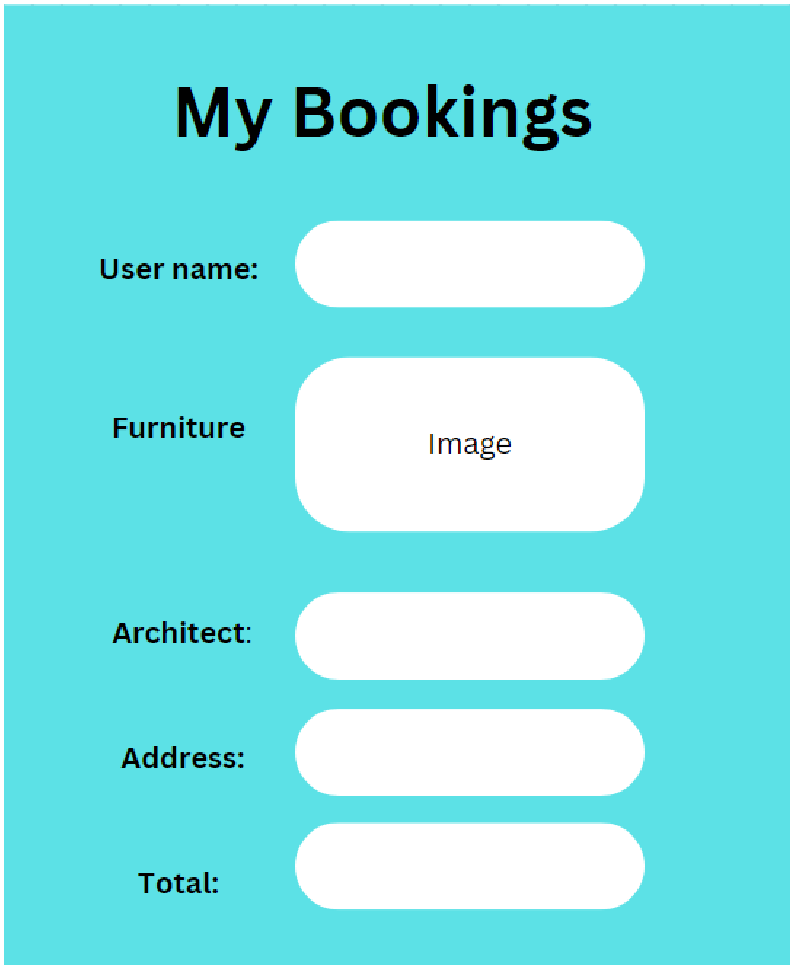
Attribute	**Datatype**	**Relationship**
User_ID	Integer(10)	PK
UserName	Varchar(30)	
Architect_Name	Varchar(10)	FK
Furniture	Varchar(10)	FK
Address	Varchar(30)	

4.2.11 Admin

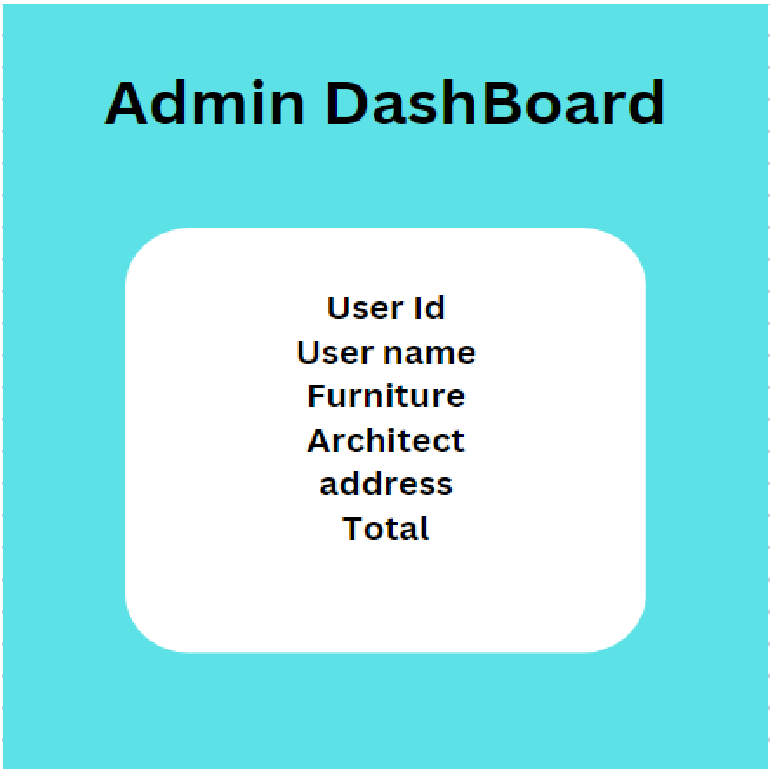
Attribute	**Datatype**	**Relationship**
Admin_ID	Integer(10)	PK
UserName	Varchar(30)	
Architect_Name	Varchar(10)	FK
Furniture	Varchar(10)	FK
Address	Varchar(30)	

4.3 User Interface Designs

4.3.1 My Bookings



4.3.2 Admin Dash Board



Admin DashBoard

User Id
User name
Furniture
Architect
address
Total

Chapter 5: Implementation and Testing

The Home Revive website offers a seamless home renovation experience for users. Upon registration, users can easily log in to explore various categories such as bedroom, kitchen, living room, bathroom, and balcony. Each category presents a photo gallery providing visual inspiration for users to envision their desired designs.

The Architect page showcases profiles of six different architects, complete with their details. Users can browse through before-and-after pictures of the architects' previous work, helping them make informed decisions about which architect to contact for their project.

To address queries and concerns, the FAQ page serves as a resource, providing clarity on common doubts users may have. Additionally, Home Revive introduces an innovative feature allowing users to purchase furniture directly from the website, aligning with their chosen home design themes.

To finalize their home improvement journey, users can seamlessly book both furniture and architects by providing necessary details. This holistic approach streamlines the renovation process, from design inspiration to professional collaboration and furniture acquisition, all within the Home Revive platform.

5.2 Coding Details and Code Efficiency

5.2.1 Code –

Register.js

```
import React , {useState, useContext} from 'react'


import { Container, Row, Col, Form, FormGroup, Button } from 'reactstrap'

import { useNavigate} from 'react-router-dom'



import { AuthContext } from '../context/AuthContext';
import { BASE_URL } from './../utils/config';



const Register = (props) => {
    const [credentials, setCredentials] = useState({
        username: '',
        email: '',
        password: ''
    });

    const [error, setError] = useState('');

    const { dispatch } = useContext(AuthContext);
    const navigate = useNavigate();

    const handleChange = e => {
        setCredentials(prev => ({ ...prev, [e.target.id.toLowerCase()]: e.target.value }));
    };

    const handleClick = async e => {
        e.preventDefault();

        try {
            const res = await fetch(`${BASE_URL}/auth/register`, {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(credentials)
            });
            const result = await res.json();

            if (!res.ok) {
                if (result && result.message) {
                    throw new Error(result.message);
                } else {
                    throw new Error('Registration failed.');
                }
            }

            dispatch({ type: 'REGISTER_SUCCESS' });
            navigate('/category');

        } catch (err) {
            setError(err.message);
        }
    };

    return (
        <section>
```

```
      <Container>
        <Row>
          <Col lg='8' className="m-auto">
            <h2>Register</h2>

            <Form onSubmit={handleClick}>
              <FormGroup>
                <input type="text" placeholder="Username" required id="username" onChange={handleChange} />
              </FormGroup>
              <FormGroup>
                <input type="email" placeholder="Email" required id="Email" onChange={handleChange} />
              </FormGroup>
              <FormGroup>
                <input type="password" placeholder="Password" required id="password" onChange={handleChange} />
              </FormGroup>

              {error && <p className="text-danger">{error}</p>}

              <Button className="btn secondary_btn auth_btn" type="submit">Create Account</Button>
            </Form>

            <button className="link-btn" onClick={() => props.onFormSwitch('login')}>Already have an account? Login.</button>
          </Col>
        </Row>
      </Container>
    </section>
  );
};

export { Register };
```

Login.js

```
import React, { useState, useContext } from 'react';
import { Container, Row, Col, Form, FormGroup, Button } from 'reactstrap';
import {  useNavigate } from 'react-router-dom';
import { AuthContext } from '../context/AuthContext';
import { BASE_URL } from '../utils/config';
//import AdminDashboard from '../pages/Admindashboard';

const Login = (props) => {
  const [credentials, setCredentials] = useState({
    email: '',
    password: '',
    isAdmin: false,
  });

  const { dispatch } = useContext(AuthContext);
  const navigate = useNavigate();
  const handleChange = (e) => {
    setCredentials((prev) => ({ ...prev, [e.target.id.toLowerCase()]: e.target.value }));
    if (e.target.id === 'email') {
      setCredentials((prev) => ({ ...prev, isAdmin: e.target.value === 'admin@example.com' }));
    }
  };

  const handleClick = async (e) => {
    e.preventDefault();

    dispatch({ type: 'LOGIN_START' });

    try {
      const res = await fetch(`${BASE_URL}/auth/login`, {
        method: 'POST', // corrected method name
        headers: {
          'Content-Type': 'application/json',
        },
        credentials: 'include',
        body: JSON.stringify(credentials),
      });

      const result = await res.json();
      if (!res.ok) {
```

```
   alert(result.message);
  }

  console.log(result.data);

  if (result.data.role === 'admin') {
   navigate('/admin');}
   else{
  dispatch({ type: 'LOGIN_SUCCESS', payload: result.data });
  navigate('/category');}
 } catch (err) {
  dispatch({ type: 'LOGIN_FAILURE', payload: err.message });
 }
};

return (
 <section>
  <Container>
   <Row>
    <Col lg="8" className="m-auto">
     <h2>Login</h2>

     <Form onSubmit={handleClick}>
      <FormGroup>
       <input type="email" placeholder="Email" required id="email" onChange={handleChange} />
```

```
              </FormGroup>
              <FormGroup>
                <input type="password" placeholder="Password" required id="password" onChange={handleChange} />
              </FormGroup>

              <Button className="btn secondary_btn auth_btn" type="submit">
                Login
              </Button>
            </Form>


            <button className="link-btn" onClick={() => props.onFormSwitch('register')}>
              Don't have an account? Register here.
            </button>
          </Col>
        </Row>
      </Container>
    </section>
  );
};

export { Login };
```

ArchitectPage.js

```
import React from 'react';
import architect1 from "../assets/architect1.jpeg";

import architect2 from "../assets/architect2.jpeg";

import architect3 from "../assets/architect3.jpg";

import architect4 from "../assets/architect4.jpeg";

import architect5 from "../assets/architect5.jpg";

import architect6 from "../assets/architect6.jpg";

import "../styles/Architect.css";


const architects = [

  {

    id: 1,

    name: 'Ramesh Kumar',

    description: 'Experienced architect specializing in modern designs.',

    contact: '8454857103',

    email: "rameshkumar@gmail.com",

    work: 'Modern Residential Designs',

    image: architect1,

  },

  {

    id: 2,

    name: 'Sanjay Puri',

    description: 'Innovative architect with a focus on sustainable and eco-friendly designs.',

    contact: '9702034925',

    email:"sanjaypuri12@gmail.com",
    work: 'Sustainable Architecture Projects',
    image: architect2,
  },
  {
    id: 3,
    name: 'Rajiv Saini',
    description: 'Passionate architect known for classical and timeless designs.',
    contact: '9004762313',
    email: "rajiv345@gmail.com",
    work: 'Classical and Timeless Designs',
    image: architect3,
  },
  {
    id: 4,
    name: 'Manoj Verma',
    description: 'Creative architect with expertise in urban planning and development.',
    contact: '7304380967',
    email: "manojverma11@gmail.com",
    work: 'Urban Planning and Development Projects',
    image: architect4,
  },
  {
    id: 5,
    name: 'Steven Alon',
    description: 'Blend artistic vision with technical expertise to design and oversee the construction of
```

```
functional, aesthetically pleasing, and safe structures..',
    contact: '8425091066',
    email: "stevenalon178@gmail.com",
    work: 'design and oversee the construction of buildings, harmonizing functionality and aesthetics.',
    image: architect5,
  },
  {
    id: 6,
    name: 'Rahul Jain',
    description: 'create visionary blueprints, well-designed structures that seamlessly integrate form and function.',
    contact: '9594430719',
    email: "rahul345@gmail.com",
    work: 'Architects envision and design structures, while also overseeing their construction to ensure realization of the intended vision.',
    image: architect6,
  },
];

const Architect = () => {
  return (
    <div>
      <h2>Meet Our Architects</h2>
      <p>Explore profiles of visionary architects and their outstanding works.</p>

      <div className="architects">
```

```
        {architects.map(architect => (
          <div className="architect-card" key={architect.id}>
            <div>
              <img src={architect.image} alt={architect.name} className="architect-image" />
              <div className="architect-details">
                <h3>{architect.name}</h3>
                <p className="work">{architect.work}</p>
                <p className="description">{architect.description}</p>
                <p className="contact">Contact: <a href={`tel:${architect.contact}`}>{architect.contact}</a></p>
                <p className="email">Email: <a href={`mailto:${architect.email}`}>{architect.email}</a></p>
              </div>
              </div>
          </div>
        ))}
      </div>
    </div>
  );
};


export default Architect;
```

Booking.js

```
import React, { useState, useContext ,useEffect} from 'react';
import { Form,FormGroup, Input, Label } from 'reactstrap';
import { useFurnitureContext } from '../pages/FurnitureContext';
import { useNavigate } from 'react-router-dom';
import { AuthContext } from '../context/AuthContext';
import { BASE_URL } from '../utils/config';
import "../styles/Booking.css";


const Booking = () => {
  const { selectedFurnitureList, clearFurnitureList } = useFurnitureContext();
  const { user } = useContext(AuthContext);
  const navigate = useNavigate();

  const architects=[
    'Ramesh Kumar',
    'Sanjay Puri',
    'Rajiv Saini',
    'Manoj Verma',
    'Steven Alon',
    'Rahul Jain',
  ]

  const [booking, setBooking] = useState({
    userId: user && user._id,
    userEmail: user && user.email,
    fullName: 'John Doe',
    phone: '8454857103',
    date: '2024-02-29',
    architect: '',
    address: 'xyz',
    furnitureList: selectedFurnitureList,
    paymentMethod: 'cash',

  });
```

```
useEffect(() => {
  const totalPrice = selectedFurnitureList.reduce((acc, furniture) => {
    return acc + parseFloat(furniture.price);
  }, 0);
  setBooking((prevData) => ({ ...prevData, furnitureData: totalPrice }));
}, [selectedFurnitureList]);


const handleChange = (e) => {
  const { id, value } = e.target;
  setBooking((prevData) => ({ ...prevData, [id]: value }));
};

const handleAddMore = () => {
    navigate('/furniture');
};

const handleClick = async (e) => {
  e.preventDefault();

  try {
    if(!user || user===undefined || user===null){
      return alert('Please sign in')
    }

    const res = await fetch(`${BASE_URL}/booking`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      credentials: 'include',
      body: JSON.stringify({ ...booking, furnitureList: selectedFurnitureList }),
    });

    const result = await res.json();

    if (!res.ok) {
      throw new Error(result.message);
    }
    navigate('/thankyou');
    clearFurnitureList();

  } catch (err) {
    alert(`Error: ${err.message}`);
  }
};

const handlePaymentSubmit = () => {
  switch (booking.paymentMethod) {
    case 'cashOnDelivery':
      alert('Order placed. Pay on delivery.');
      break;
    case 'googlePay':
      alert('Redirecting to Google Pay...');
      break;
    case 'netBanking':
      alert('Redirecting to Net Banking...');
      break;
    default:
      break;
  }
};

const handlePaymentChange = (e) => {
  const { value } = e.target;
  setBooking({ ...booking, paymentMethod: value });
};

return (
  <div className="booking__form">
    <h1>Effortless Booking for Exceptional Service: Secure Your Home Revive Appointment Now!</h1>
    <Form className="booking__info-form" onSubmit={handleClick}>
      <h6>Enter Your Name </h6>
      <FormGroup className="d-flex align-itens-center gap-4 ">
        <Input
          type="text"
          placeholder="Name"
          id="fullName"
          required
          onChange={handleChange}
        />
      </FormGroup>

      <h6>Enter Your Contact Details </h6>
      <FormGroup className="d-flex align-itens-center gap-4 ">
        <Input
          type="tel"
          placeholder="Phone Number"
          id="phone"
          required
          onChange={handleChange}
        />
      </FormGroup>
      <h6>Select A date</h6>
      <FormGroup className="d-flex align-itens-center gap-4">
        <Input
          type="date"
          placeholder="checkinDate"
          id="date"
          required
          onChange={handleChange}
        />
      </FormGroup>
      <h6>Enter architect Details</h6>
      <FormGroup>
        <Input
          type="select"
          placeholder="Name of the Architect"
          id="architect"
          required
          onChange={handleChange}
          >
          <option value="" disabled selected>Select an Architect</option>
          {architects.map((architect, index) => (
            <option key={index} value={architect}>
              {architect}
            </option>
          ))}
        </Input>
```

```
</FormGroup>
<h6>Enter Complete Address</h6>
<FormGroup>
  <Input
    type="text"
    placeholder="Address"
    id="address"
    required
    onChange={handleChange}
  />
</FormGroup>


{selectedFurnitureList.length > 0 && (
  <div>
    <h3>Selected Furniture:</h3>
    {selectedFurnitureList.map((furniture, index) => (
      <div key={index}>
        <img src={furniture.image} alt={furniture.name} />
        <p>{furniture.name}</p>
        <p>{furniture.Des}</p>
        <p>{furniture.price}</p>
      </div>
    ))}
  <div>
  <h6>Total Amount: {booking.furnitureData}</h6>

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

```
<button type="button" onClick={handleAddMore} className="add-more-button">
  Add More
</button>



<div className="payment-section mt-4">
  <div>
    <h5>Payment Options</h5>
    {/* ... (rest of the payment options) */}
    <FormGroup check>
      <Input
        name="radio1"
        type="radio"
        value="cash"
        checked={booking.paymentMethod === 'cash'}
        onChange={handlePaymentChange}
      />
      {' '}
      <Label check>
        Cash On Delivery
      </Label>
    </FormGroup>
    <FormGroup check>
      <Input
        name="radio1"
        type="radio"
        value="googlePay"
        checked={booking.paymentMethod === 'googlePay'}
        onChange={handlePaymentChange}
```

```
            />
            {' '}
            <Label check>
              Google Pay
            </Label>
          </FormGroup>
          <FormGroup
            check
            disabled
          >
            <Input
              name="radio1"
              type="radio"
              value="credit"
              checked={booking.paymentMethod === 'credit'}
              onChange={handlePaymentChange}
            />
            {' '}
            <Label check>
              Credit Card
            </Label>
          </FormGroup>

          <button onClick={handlePaymentSubmit}>Proceed to Payment</button>

        </div>
      </div>
    </Form>
  </div>
);
```

```
};

export default Booking;
```

AuthController.js

```
import User from "../models/user.js";
import bcrypt from "bcryptjs"
import jwt from "jsonwebtoken";



//user registration
export const register = async(req,res)=>{
  try {

    //hashing password
    const salt = bcrypt.genSaltSync(10)
    const hash = bcrypt.hashSync(req.body.password, salt)

    const newUser = new User({
      username: req.body.username,
      email: req.body.email,
      password: hash
    })

    await newUser.save()
```

```
        res.status(200).json({success:true, message:'Successfully Created'})


    } catch (err) {
        console.log(err)
        res.status(500).json({success:false, message:'Failed to create. Try again'})
    }
};


//user login
export const login = async(req,res)=>{


    const email = req.body.email


    try {


        const user = await User.findOne({email})


        // if user doesn't exist
        if(!user){
            return res.status(404).json({success:false, message:"User not found."});
        }




        // if user is exist then check the password or compare the password
        const checkCorrectPassword = await bcrypt.compare(req.body.password, user.password)
        //if password is Incorrect
        if(!checkCorrectPassword){
            return res.status(401).json({success:false, message:"Incorrect email or password"})
        }

        const {password, role, ...rest} = user._doc

        //create jwt token
        const token = jwt.sign(
            { id: user._id, role: user.role },
            process.env.JWT_SECRET_KEY,
            { expiresIn: "15d" }
        );


        // set token in the browser cookies and send the response to the client
        res.cookie('accessToken', token, {
            httpOnly: true,
            expires:new Date(Date.now() + 15 * 24 * 60 * 60 * 1000),
        }).status(200).json({token , data:{...rest}, role,})

    } catch (err) {
        console.error(err);
        res.status(500).json({success:false, message:"Failed to login"})
    }
}
```

UserController.js

```
import User from '../models/user.js'

//create new User

export const createUser = async (req,res)=>{

    const newUser = new User(req.body);

    try{

        const savedUser = await newUser.save();

        res.status(200).json({success:true, message:'Successfully created',

        data:savedUser});

    } catch (err) {

        res.status(500).json({success:false, message:'failed to Create. Try again'})

    }

} ;

//update User

export const updateUser = async (req,res) =>{

    const id = req.params.id;

    try {

        const updatedUser = await User.findByIdAndUpdate(id, {

            $set: req.body

        }, {new:true})

        res.status(200).json({success:true, message:'Successfully Updated',

        data: updatedUser,
    });



    } catch (err) {
        res.status(500).json({success:false, message:'failed to update'
        });
    }
};
//delete User
export const deleteUser = async (req, res) => {
    const id = req.params.id;
    try {
        await User.findByIdAndDelete(id);
        res.status(200).json({
            success: true,
            message: "Sucessfully deleted",
        });
    } catch (err) {
        res.status(500).json({
            success: false,
            message: "failed to delete",
        });
    }
};
```

```
//get SingleUser
export const getSingleUser = async (req, res) => {
    const id = req.params.id;
    try {
        const user = await User.findById(id);
        res.status(200).json({
            success: true,
            message: "Sucessful",
            data: user,
        });
    } catch (err) {
        res.status(404).json({
            success: false,
            message: "not found",
        });
    }
};

//getAll tour
export const getAllUser = async (req, res) => {

    try {
        const users = await  User.find({});
        res.status(200).json({
            success: true,
           message: "Sucessful",
            data: users,
        })
    } catch (err) {
        res.status(404).json({
            success: false,
            message: "not found",
        });
    }
};
```

Route Booking.js

```
import express from 'express'
import { verifyAdmin, verifyUser } from '../utils/verifyToken.js';
import { createBooking, getAllBooking, getBooking, getBookingForUser } from
'../Controllers/BookingController.js';
const router = express.Router()
router.post('/', verifyUser, createBooking );
router.get("/:id", verifyUser, getBooking );
router.get('/', verifyAdmin, getAllBooking );
router.get('/:id', verifyUser, getBooking);
router.get('/user/:userId', verifyUser, getBookingForUser);


export default router
```

Index.js

```
import express from "express";
import dotenv from "dotenv";
import mongoose from "mongoose";
import cors from "cors";
import cookieParser from "cookie-parser";
import userRoute from "./Routes/user.js";
import authRoute from "./Routes/auth.js";
import bookingRoute from "./Routes/booking.js";
dotenv.config()
const app = express()
const port = process.env.PORT || 8000;
const corsOptions = {
    origin:true,
    credentials:true
}
//database connection
mongoose.set('strictQuery',false);
const connect = async () => {
    try {
        await mongoose.connect(process.env.MONGO_URL,{
        })
        console.log('MongoDB database connected');

    } catch (err) {
        console.log('MongoDB databse connection failed ')
    }
};
//middleware
app.use(express.json());
app.use(cors(corsOptions));
app.use(cookieParser());
app.use('/api/v1/auth', authRoute);
app.use('/api/v1/user', userRoute);
app.use('/api/v1/booking', bookingRoute);
app.listen(port, ()=>{
    connect();
    console.log("server listening on port" , port);
})
```

App.js

```
import React, { useContext } from "react";
import "./App.css";
import Navbar from "./components/Navbar";
import Footer from "./components/Footer";
import Home from "./pages/Home";
import Category from "./pages/Category";
import LivingRoomPage from "./pages/LivingRoomPage";
import Kitchen from "./pages/Kitchen";
import Bathroom from "./pages/Bathroom";
import Bedroom from "./pages/Bedroom";
import Balcony from "./pages/Balcony";
import Architect from "./pages/Architect";
import Photos from "./pages/Photos";
import Furniture from "./pages/Furniture";
import FurnitureData from "./pages/FurnitureData";
import FAQ from "./pages/FAQ";
import Booking from "./pages/Booking";
import Thankyou from "./pages/Thankyou";
import AboutPage from "./pages/AboutPage";
import UserBookings from './pages/UserBookings';
import Admin from "./pages/Admindashboard";
import { FurnitureProvider } from './pages/FurnitureContext';
import { BrowserRouter , Route, Routes } from "react-router-dom";
import { AuthContext } from "./context/AuthContext";
function App() {
  const { user } = useContext(AuthContext);
  return (
    <div className="App">
      <BrowserRouter>
        <Navbar />
        <FurnitureProvider>
        <Routes>
          <Route path="/" exact element={<Home />} />
          <Route path="/home" exact element={<Home />}   />
          <Route path="/category" exact element={<Category />} />
          <Route path="/livingroompage" exact element={<LivingRoomPage />} />
          <Route path="/kitchen" exact element={<Kitchen />} />
          <Route path="/bathroom" exact element={<Bathroom />} />
          <Route path="/bedroom" exact element={<Bedroom />} />
          <Route path="/balcony" exact element={<Balcony />} />
          <Route path="/architect" exact element={<Architect/>} />
          <Route path="/photos" exact element={<Photos/>} />
          <Route path="/furniture" exact element={<Furniture/>} />
          <Route path="/furnituredata/:id" exact element={<FurnitureData/>} />
          <Route path="/faq" exact element={<FAQ/>} />
          <Route path="/booking" exact element={<Booking/>} />
          <Route path="/user/bookings" exact element={<UserBookings />} />
          <Route path="/thankyou" exact element={<Thankyou/>} />
          <Route path="/aboutpage" exact element={<AboutPage/>} />
          {user?.email === "admin@gmail.com" &&  <Route path="/admin" exact element={<Admin/>} />}
            </Routes>
        </FurnitureProvider>
        <Footer />
      </BrowserRouter>
      </div>
  );
}

export default App;
```

5.2.2 COCOMO

COCOMO, or the Constructive Cost Model, stands as a prominent tool in software engineering, crafted by Barry Boehm during the late 1970s. Widely embraced in the industry, it facilitates the estimation of cost, effort, and schedule for software projects. This model empowers project managers and developers with valuable insights, aiding in strategic decisions regarding resource allocation, budgeting, and project timelines.

Barry Boehm's classification system categorizes software development projects into three types: organic, semidetached, and embedded, based on their inherent development complexity. This classification allows for a nuanced understanding of the project dynamics, providing a tailored approach to estimation and planning according to the specific challenges posed by each category.

Organic: COCOMO is well-suited for projects characterized by well-defined requirements, a seasoned development team, and a stable environment. It proves valuable for project managers by aiding in resource allocation and scheduling decisions throughout the software development lifecycle.

Semi-detached COCOMO finds its application in medium-sized projects featuring a blend of newly developed and reused components. This model delivers more accurate estimates by addressing the intricacies of integrating both new and existing elements, empowering project managers to make well-informed decisions regarding resource allocation and scheduling in projects with varying levels of component reuse.

Embedded COCOMO represents a specialized variant tailored for projects tightly integrating software with hardware, commonly observed in devices or machinery. This model considers factors such as processing speed and memory constraints, enabling project managers to gauge the time and effort required for developing software in such systems. It provides a targeted approach for decision-making in projects where software intricately interacts with hardware components.

The formula used for the cost estimation for the COCOMO model is given below:

Effort = a*(KLOC)b [KLOC=lines of code(thousands)]

Time = c*(Effort)d

Staffing = Effort/Time

Type	a	b	c	d	KLOC
Semi - Detached	3	1.12	2.5	0.35	2.774

Effort (person-months)	**9.2920**
Time (months)	**5.4547**
Staffing	**1.7034**

5.3 Testing Approaches

Testing is a systematic process of evaluating a software application or system to identify defects, errors, or bugs. Its primary goal is to ensure that the software meets specified requirements, functions correctly, and performs reliably under various conditions. Testing helps make sure that the software works well and meets the needs of its users.

Unit testing: Validating that each software unit performs as expected. A unit is the smallest testable component of an application. Instead of testing the whole program at once, developers focus on testing each component separately. This helps catch any errors or bugs early on, making it easier to fix them before they cause bigger problems in the overall software.

5.4 Test Cases

5.4.1 Login Test Cases

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Enter Valid Email & Password	1)Enter valid Email 2)Enter Valid Password	Email:- soni11t@g mail .com Password:- ad123	Successfully logged in into User Account	Successfully logged in into User Account	PASS
2	Enter Valid Email & Invalid Password	1)Enter valid Email 2)Enter invalid Password	Email:- soni11t@g mail .com Password:- naks123	Alert with Incorrect Email or Password	Alert with Incorrect Email or Password	PASS
3	Enter invalid Email & valid Password	1)Enter invalid Email 2)Enter valid Password	Email:- test@gmail.com Password:- test@123	Alert with Incorrect Email or Password	Alert with Incorrect Email or Password	PASS
4	Enter Credentials for user who does not exist	1)Enter Email 2)Enter Password	Email:- Invalidemail @gmail .com Password:- invalidpass @123	Alert with User not Found	Alert with User not Found	PASS

Table 5.2 Login Test Cases

5.4.2 Register Test Cases

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Enter username, email, password	1.Enter username 2. Enter	Naksha Naksha11@	Redirect to category page and success	Redirected Category page and 'Sign up	PASS

		valid email 4.Enter valid password	gmail.com naksha11	message should be displayed as ‘Sign up successfully ”.	successfully’ message is seen.	
2	Enter invalid Email	1.Enter invalid Email	Jwkjgfnakedb.com	Error message should be displayed saying ‘Please entry @in your email address.’	Error message should be displayed saying ‘Please entry @in your email address.’ Is seen	PASS
3	Enter a strong password	Enter a strong password	1234	Error message should display saying ”Please gave a strong password”.	Error message should display saying ”Please gave a strong password”. Is seen	PASS
4	Enter already existing email id	1.Enter a email id that is already registered	Naksha11@gmail.com	Error message should be displayed saying ‘The email address is already in use’	Error message should be displayed saying ‘The email address is already in use’ Is seen.	PASS
5	Keep any of the field empty	1. Keep any of the field empty	Name: Naksha Email: naksha11@gmail.com Username: empty Password : naksha1234	Error message should be displayed saying ‘Username cannot be empty’	Error message should be displayed saying ‘Username cannot be empty’ is seen.	PASS

Table 5.2 Register test cases

5.4.3 Booking Test Cases

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Fill all the details required for booking [Considering user is logged in]	Fill all the details	Enter Your Name : Naksha Shetty Enter Your Contact Details: 8454857103 Select a Date: 19-03-2024 Enter Architect Detail: Ramesh Kumar Enter Complete Address Select Furniture: Sofa	Redirected to Thank YOU Page	Redirected to Thank You page	PASS

Table 5.5 Booking Module Test Cases

Chapter 6: Results and Discussions

A testing report provides a thorough overview of the testing process conducted on a software application or website. It offers insights into the testing procedures and outcomes, aiding in the comprehension of the software's performance.

- **Home Page Module:**

The Home Page Module is the main section of a website where visitors first land. It typically includes essential information, navigation links, and features to guide users to other parts of the site. It serves as a gateway to the website's content and often showcases key offerings or promotions to engage visitors.

- **Register / Login Module:-**

The user authentication module enables users to either create a new account or log in to their existing one.

- **Category**

The category navigation module organizes content into distinct categories such as bedroom, kitchen, living room, bathroom, and balcony, facilitating easy exploration for users seeking inspiration or specific design ideas.

- **Architect Profile Module:**

The architect profile module showcases detailed profiles of architects, including their portfolios and previous work, assisting users in making informed decisions when selecting an architect for their project.

- **Photos**

The Before & After showcase module displays comparative images of spaces before and after renovation, providing users with visual evidence of the transformation achieved by the renovation services offered on the website.

- **Furniture module**

The furniture catalog module presents a comprehensive collection of furniture items, ranging from sofas and tables to chairs and cabinets, allowing users to browse and select pieces that complement their design preferences and enhance their living spaces.

- **FAQ module**

The FAQ module serves as a repository of commonly asked questions and their answers, providing users with quick access to information about the website's services, policies, and processes, thereby addressing their queries and concerns efficiently.

- **Booking module**

The Booking Module on Home Revive allows users to securely provide personal details such as name, contact number, and address. Additionally, users can specify preferred dates for the renovation project, select furniture items, and choose the architect they wish to collaborate with. This comprehensive feature ensures efficient coordination and customization of services, facilitating a smooth and tailored home renovation experience for users.

- **MyBooking module :-**

The user profile module allows users to manage their bookings and account details, providing a personalized experience.

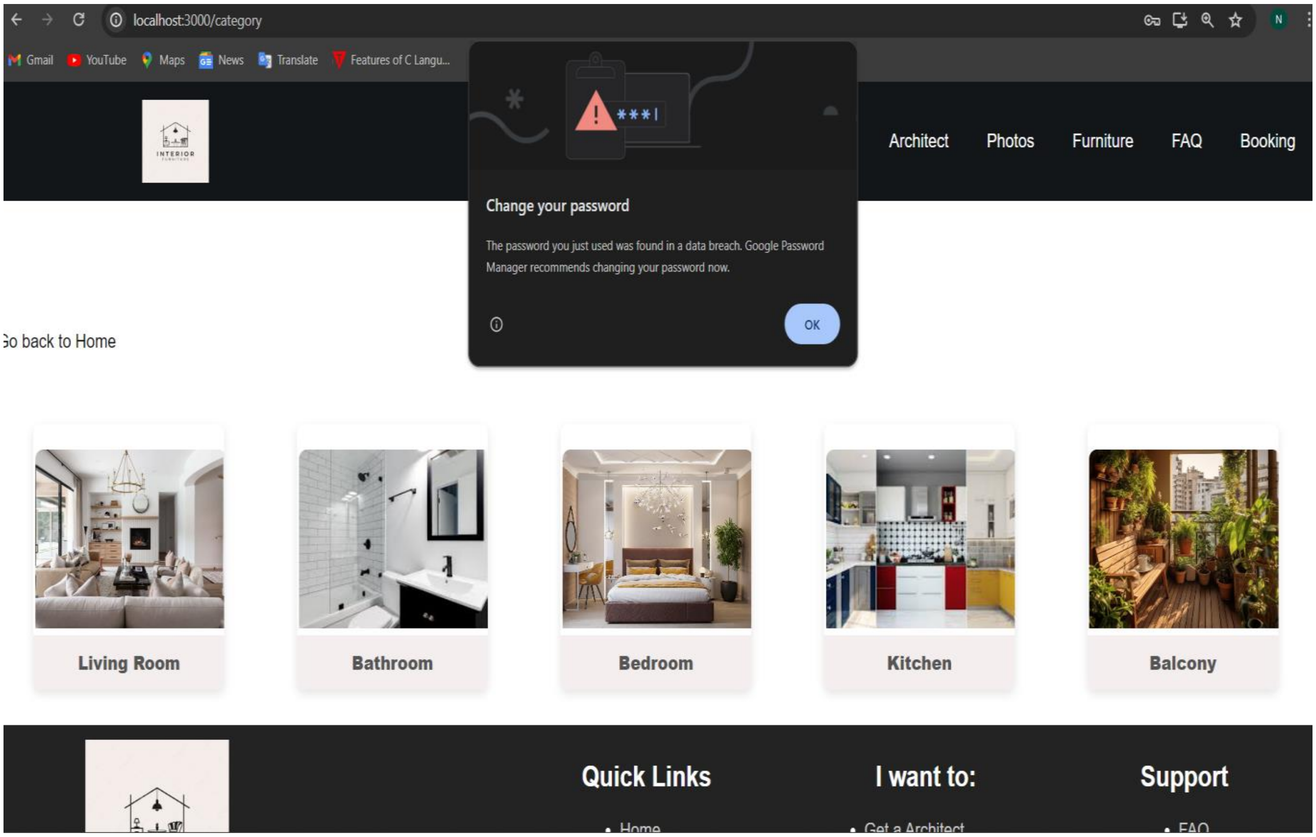
6.2. User Documentation

Welcome to Home Revive! Our website offers a user-friendly platform for seamless home renovations. Once registered, users can explore various design categories like bedrooms, kitchens, and bathrooms, each accompanied by inspiring photo galleries. Additionally, the Architect page features profiles of six skilled architects, showcasing their previous work with before-and-after pictures, aiding users in selecting the right professional for their project. For addressing queries, our FAQ page provides clarity on common doubts. Excitingly, users can now purchase furniture directly from the website, matching their chosen design themes. To finalize their renovation journey, users can easily book both furniture and architects, streamlining the entire process from design inspiration to professional collaboration and furniture acquisition, all within the Home Revive platform. We're committed to ensuring a seamless and enjoyable experience for all our users. Happy renovating!

Home Revive provides the following functions :-

Register module-

If the user's password is too short or easily guessable, a notification will prompt them to change their password for security reasons.



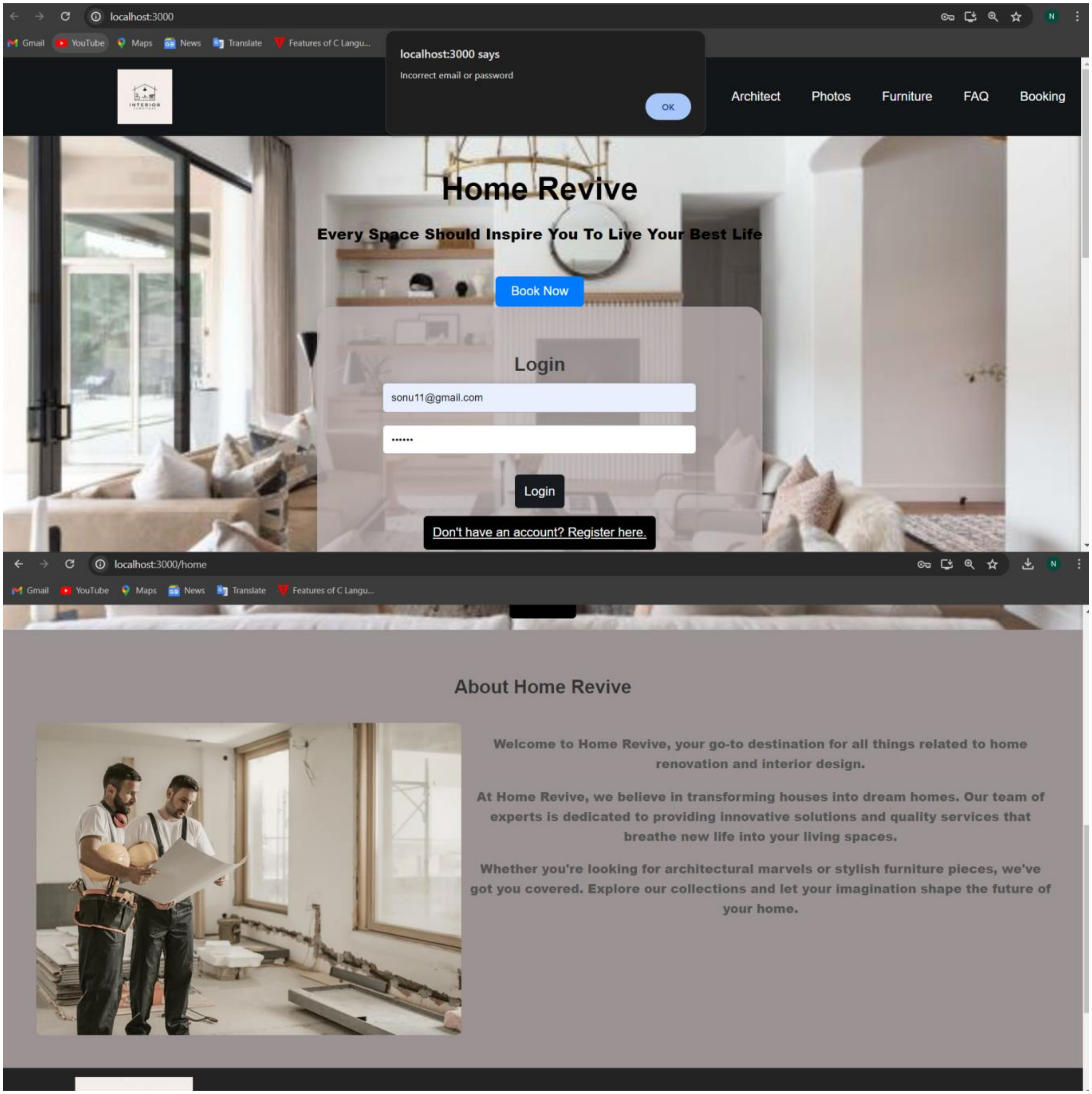
Change your password

The password you just used was found in a data breach. Google Password Manager recommends changing your password now.

6.2.1 Register page

Login module-

In case of an incorrect password entry by the user, a notification will appear notifying them of an error, stating "Incorrect email or password".



localhost:3000 says

Incorrect email or password

Home Revive

Every Space Should Inspire You To Live Your Best Life

Login

sonu11@gmail.com

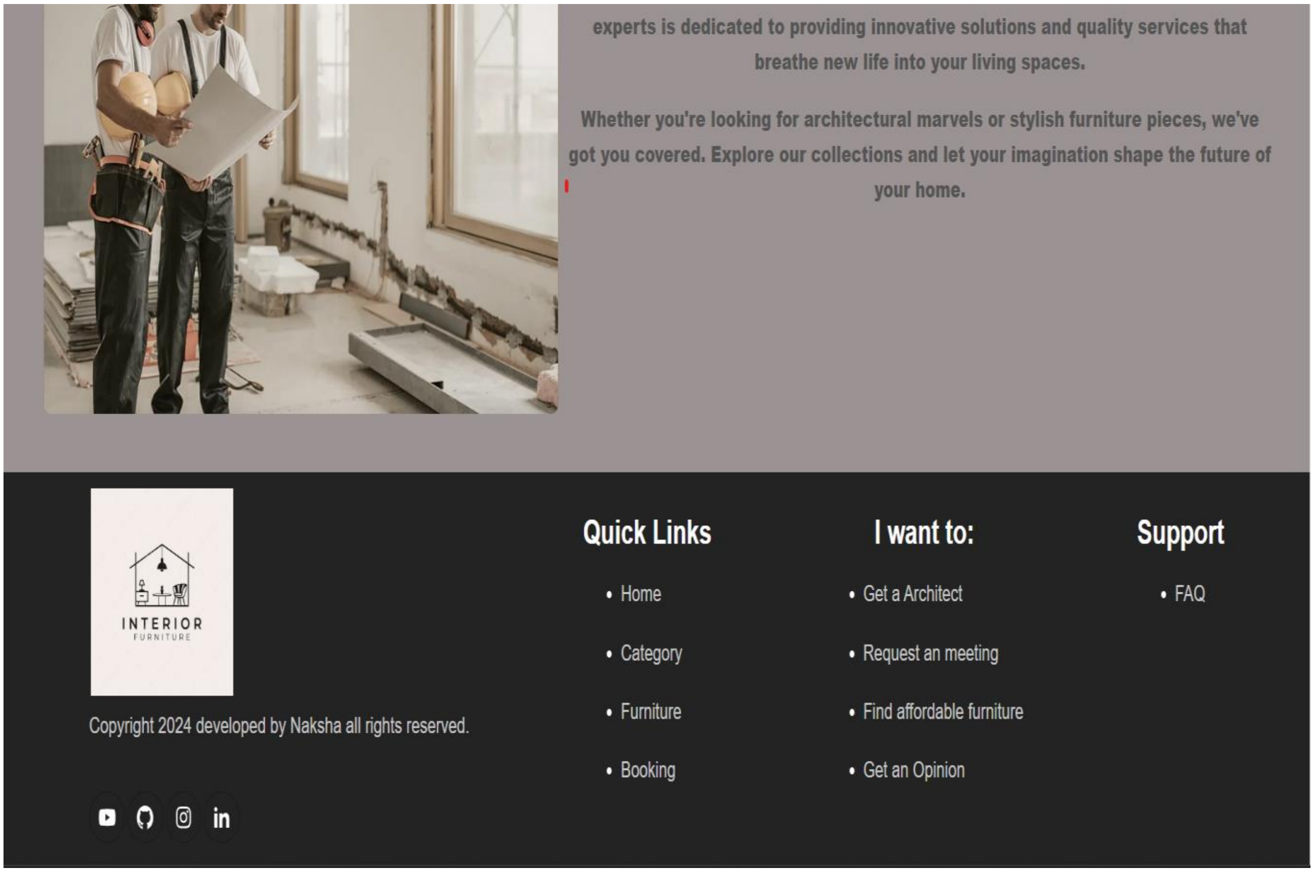
••••••

About Home Revive

Welcome to Home Revive, your go-to destination for all things related to home renovation and interior design.

At Home Revive, we believe in transforming houses into dream homes. Our team of experts is dedicated to providing innovative solutions and quality services that breathe new life into your living spaces.

Whether you're looking for architectural marvels or stylish furniture pieces, we've got you covered. Explore our collections and let your imagination shape the future of your home.



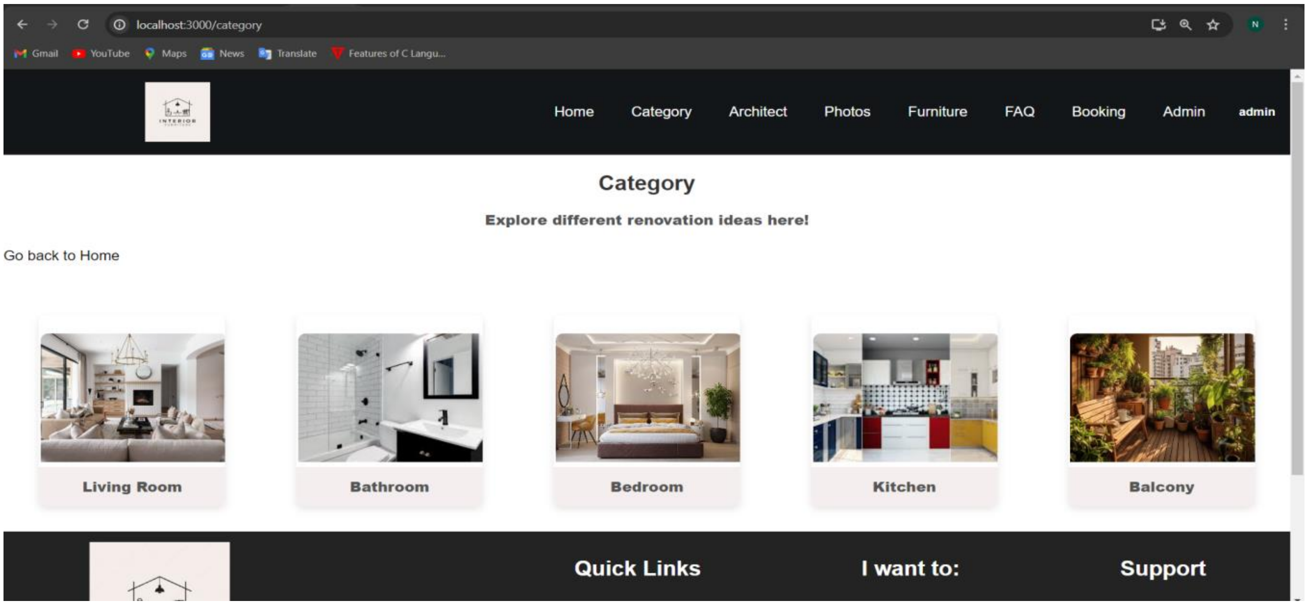
experts is dedicated to providing innovative solutions and quality services that breathe new life into your living spaces.

Whether you're looking for architectural marvels or stylish furniture pieces, we've got you covered. Explore our collections and let your imagination shape the future of your home.

Copyright 2024 developed by Naksha all rights reserved.

6.2.2 Login Page

Category Module- The Category Module in HomeRevive helps users find different design ideas for specific rooms in their home, like the bedroom,living room,bathroom,balcony or kitchen. It makes it easy to explore various styles, such as modern or vintage, so users can pick the one they like best for their space.

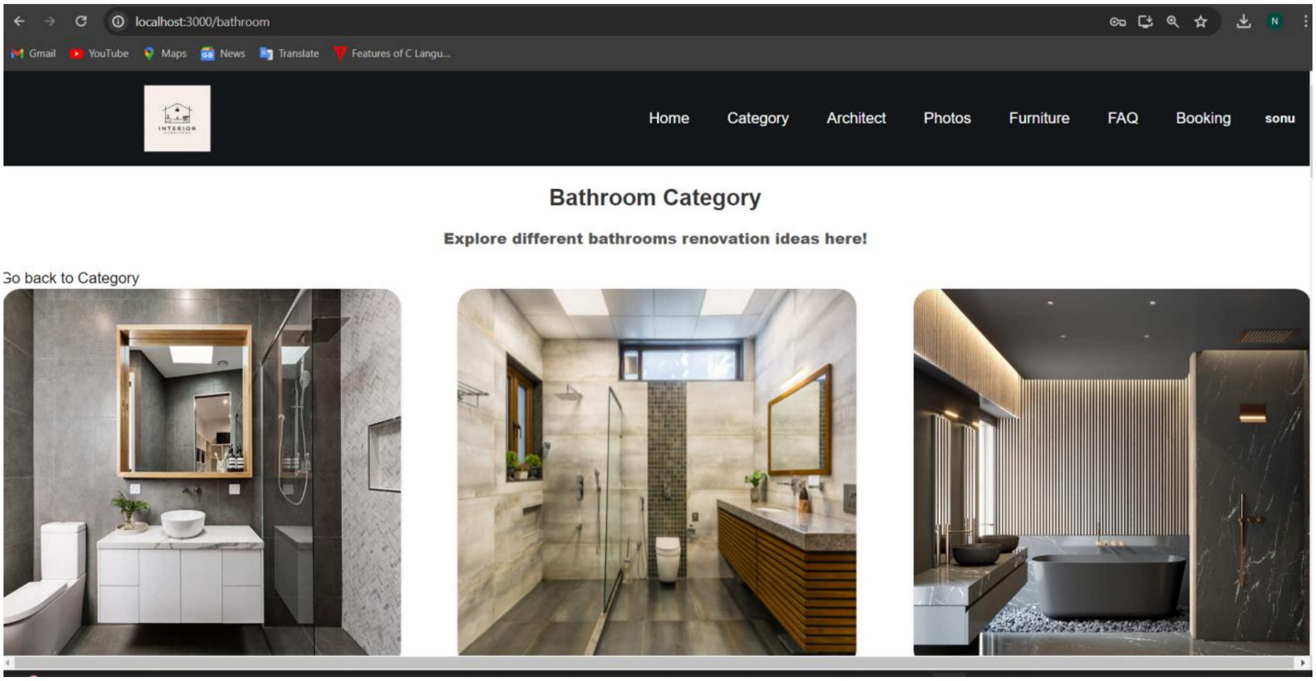


localhost:3000/category

Category

Explore different renovation ideas here!

6.2.3 Category Page



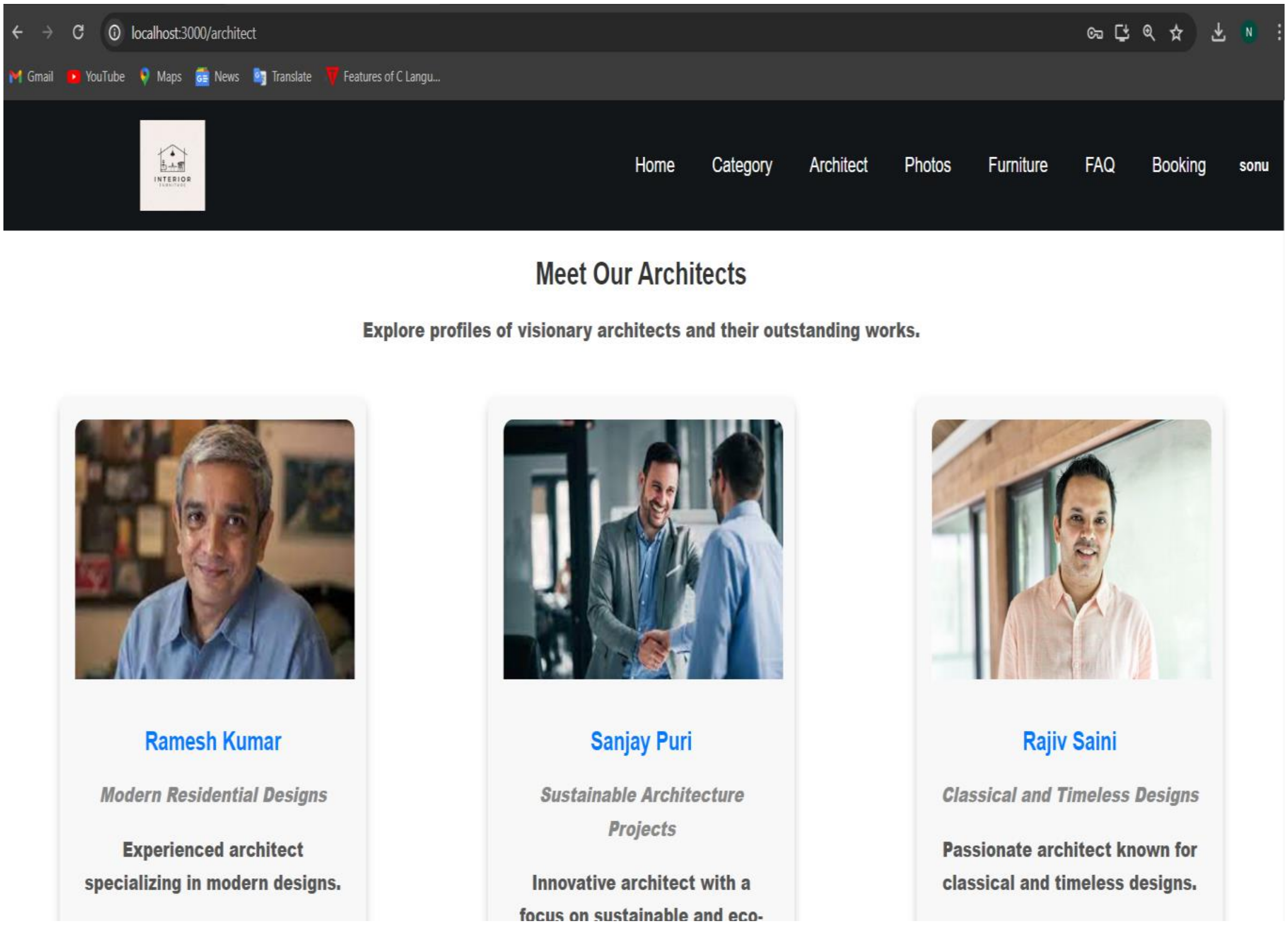
localhost:3000/bathroom

Bathroom Category

Explore different bathrooms renovation ideas here!

6.2.3 Category Page

Architect Module- In HomeRevive, the Architect Module connects users with experienced architects who can turn their design visions into reality. Users can collaborate with architects to discuss their ideas and receive expert guidance for personalized home renovation solutions.



Meet Our Architects

Explore profiles of visionary architects and their outstanding works.

Ramesh Kumar

Modern Residential Designs

Experienced architect specializing in modern designs.

Sanjay Puri

Sustainable Architecture Projects

Innovative architect with a focus on sustainable and eco-

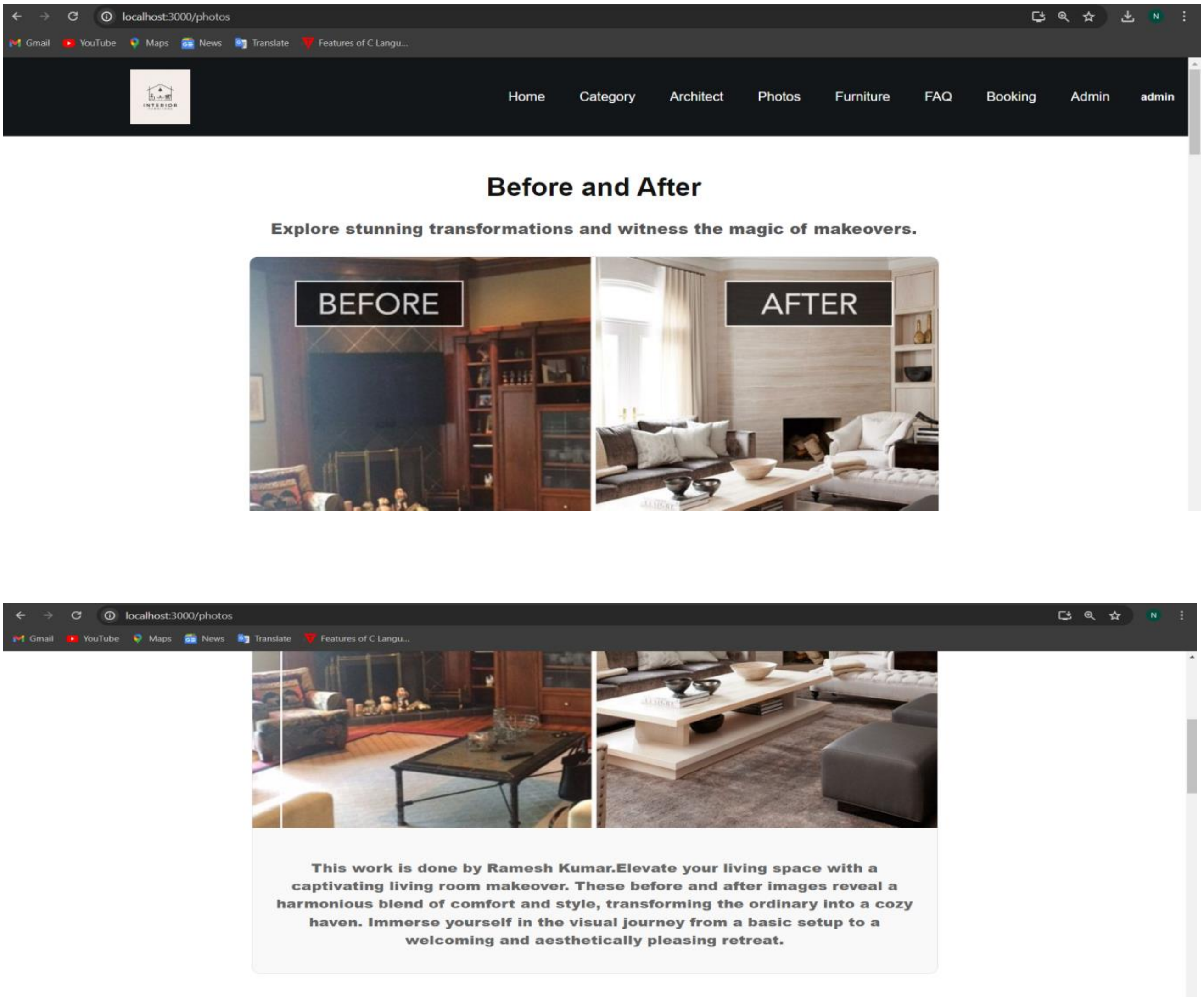
Rajiv Saini

Classical and Timeless Designs

Passionate architect known for classical and timeless designs.

6.2.4 Architcet Page

Photo Module- The Photo Module in HomeRevive showcases completed renovation projects, offering users inspiration and insights into design transformations. Users can browse before-and-after images to visualize potential outcomes and make informed decisions for their own home improvement projects.



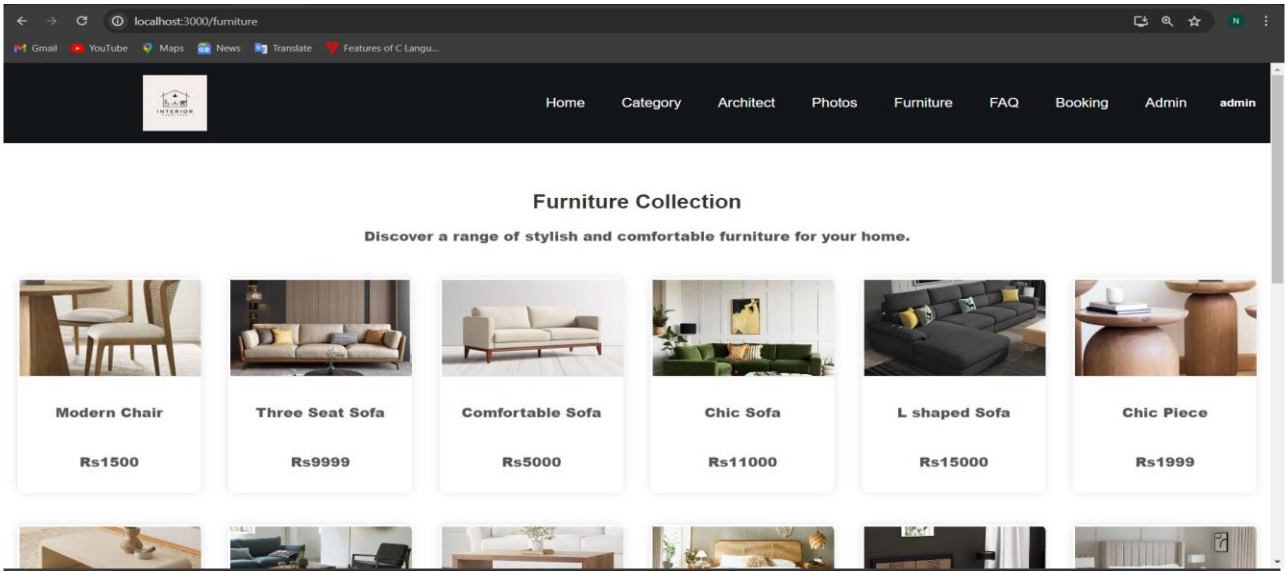
Before and After

Explore stunning transformations and witness the magic of makeovers.

This work is done by Ramesh Kumar.Elevate your living space with a captivating living room makeover. These before and after images reveal a harmonious blend of comfort and style, transforming the ordinary into a cozy haven. Immerse yourself in the visual journey from a basic setup to a welcoming and aesthetically pleasing retreat.

6.2.5 Photo Page

Furniture Module - Within HomeRevive, the Furniture Module presents a curated catalog of furnishings, ranging from sofas to cabinets, aligned with diverse design styles. Users can effortlessly explore and select furniture items that complement their preferred aesthetic, streamlining the process of furnishing their homes.



Furniture Collection

Discover a range of stylish and comfortable furniture for your home.

Modern Chair
Rs1500

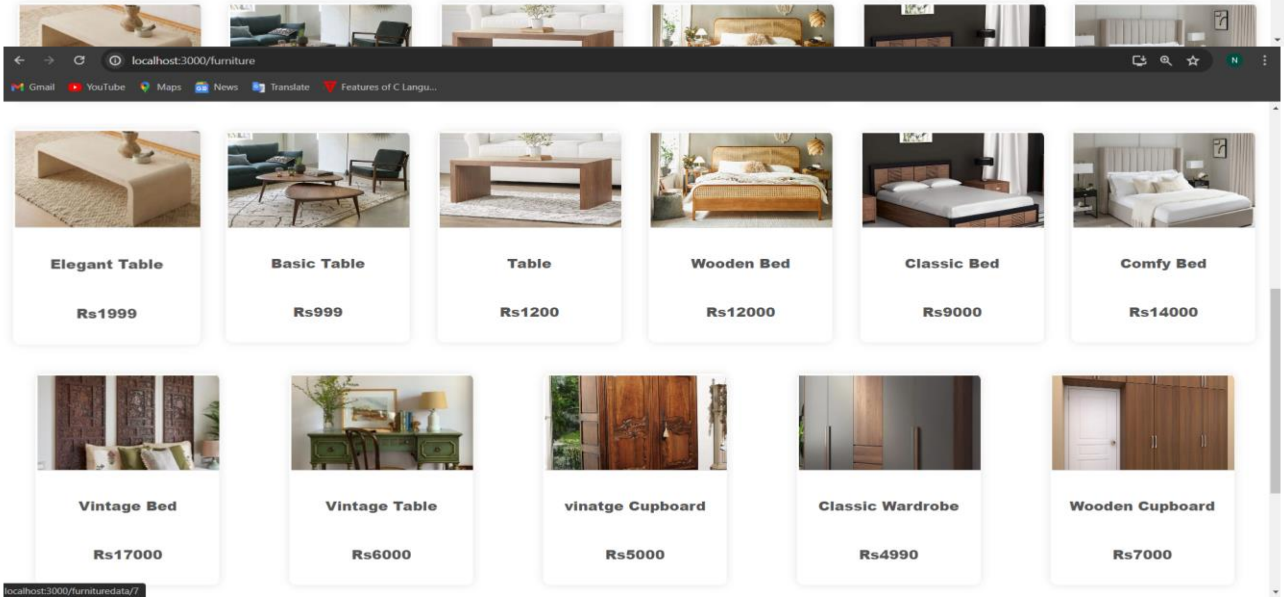
Three Seat Sofa
Rs9999

Comfortable Sofa
Rs5000

Chic Sofa
Rs11000

L shaped Sofa
Rs15000

Chic Piece
Rs1999



Elegant Table
Rs1999

Basic Table
Rs999

Table
Rs1200

Wooden Bed
Rs12000

Classic Bed
Rs9000

Comfy Bed
Rs14000

Vintage Bed
Rs17000

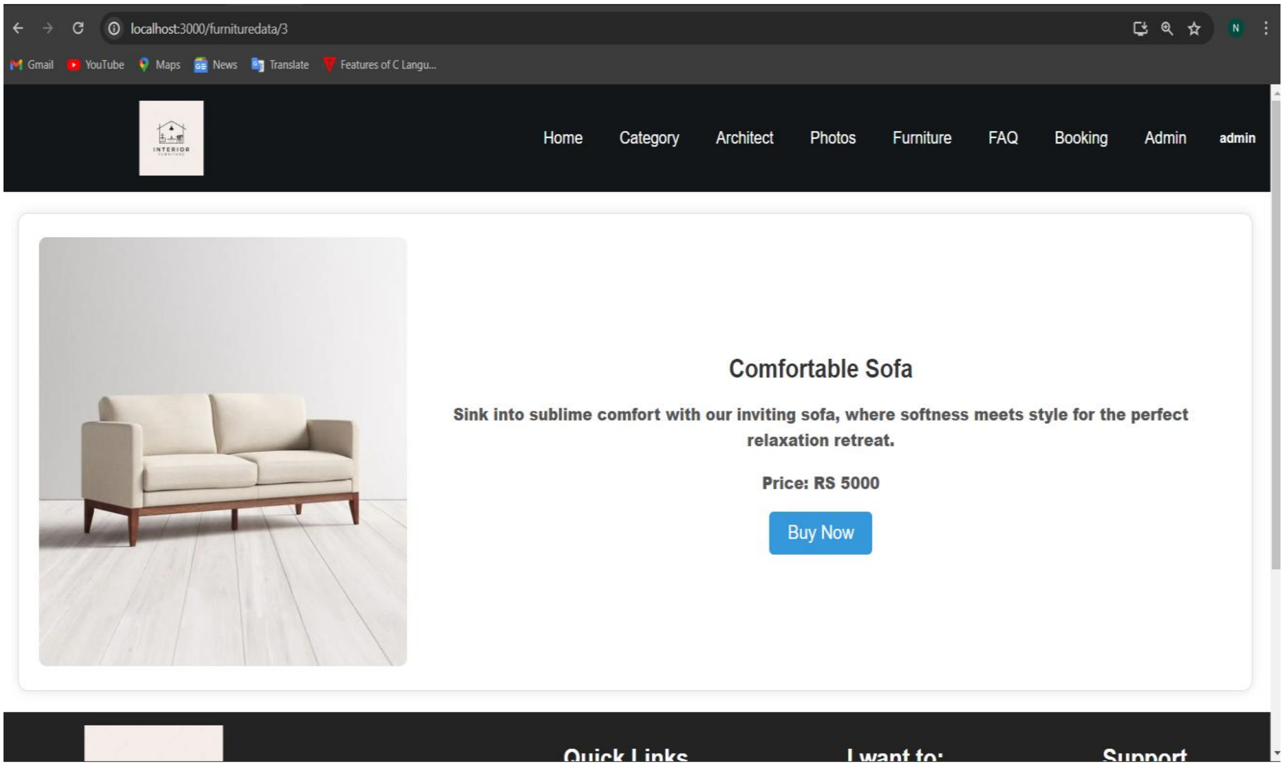
Vintage Table
Rs6000

vinatge Cupboard
Rs5000

Classic Wardrobe
Rs4990

Wooden Cupboard
Rs7000

localhost:3000/furnituredata/7



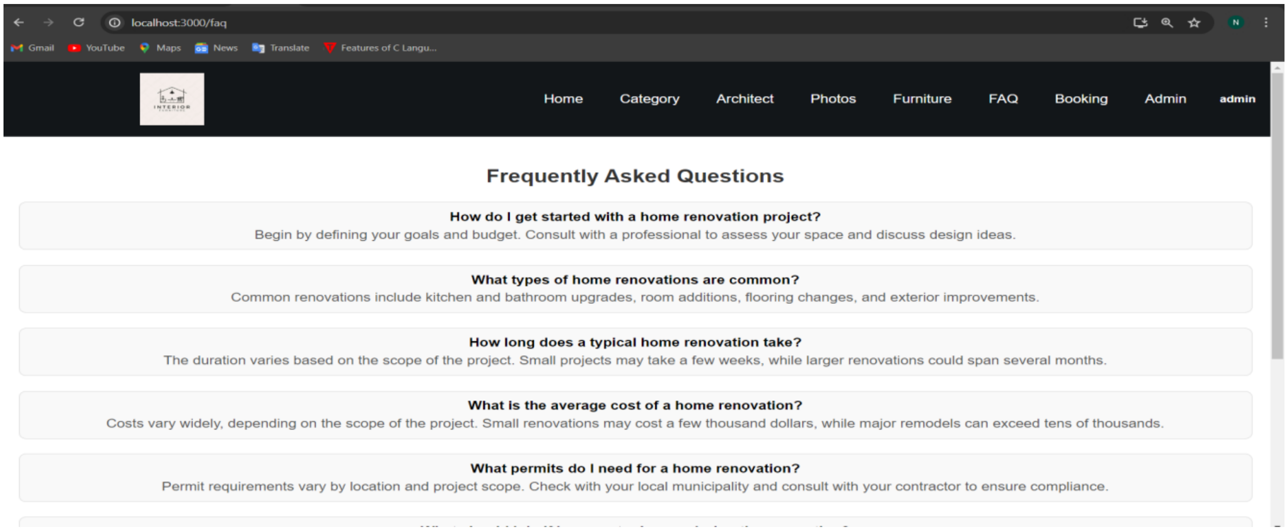
Comfortable Sofa

Sink into sublime comfort with our inviting sofa, where softness meets style for the perfect relaxation retreat.

Price: RS 5000

6.2.6 Furniture Page

FAQ- HomeRevive's FAQ section offers quick answers to common user inquiries, providing clarity on services, policies, and processes. Users can easily find relevant information to address their questions and make informed decisions about their home renovation journey.



Frequently Asked Questions

How do I get started with a home renovation project?
Begin by defining your goals and budget. Consult with a professional to assess your space and discuss design ideas.

What types of home renovations are common?
Common renovations include kitchen and bathroom upgrades, room additions, flooring changes, and exterior improvements.

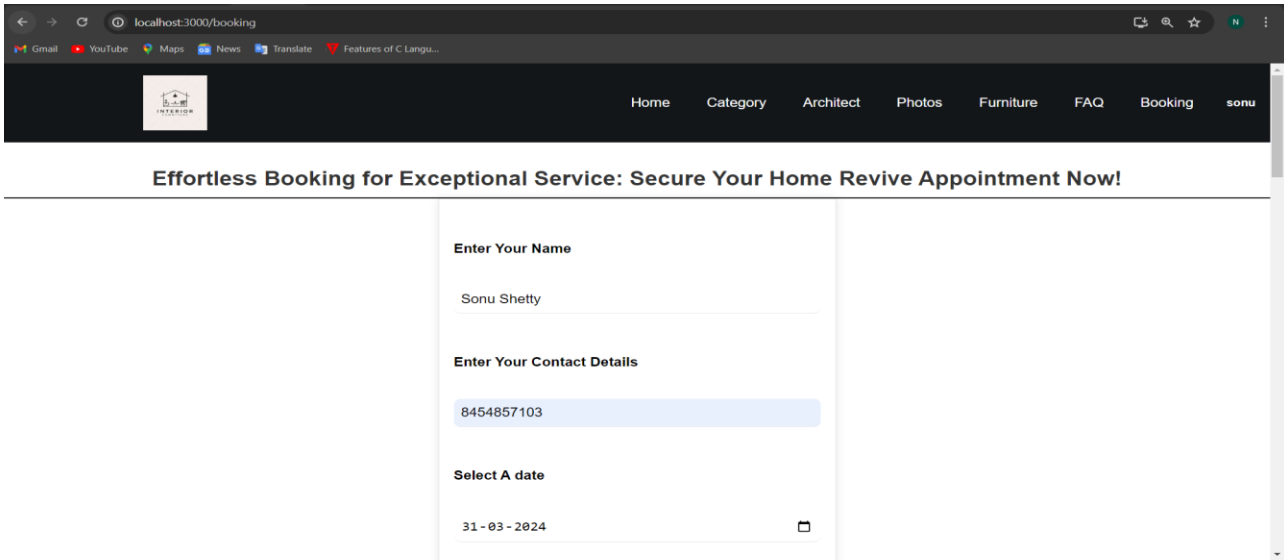
How long does a typical home renovation take?
The duration varies based on the scope of the project. Small projects may take a few weeks, while larger renovations could span several months.

What is the average cost of a home renovation?
Costs vary widely, depending on the scope of the project. Small renovations may cost a few thousand dollars, while major remodels can exceed tens of thousands.

What permits do I need for a home renovation?
Permit requirements vary by location and project scope. Check with your local municipality and consult with your contractor to ensure compliance.

6.2.7 FAQ Page

Booking Module- The Booking Module in HomeRevive streamlines the process for users to schedule renovation services, allowing them to specify project details and preferred dates with ease. Users can efficiently communicate their requirements, ensuring seamless coordination between them and service providers for a hassle-free experience.



Effortless Booking for Exceptional Service: Secure Your Home Revive Appointment Now!

Enter Your Name

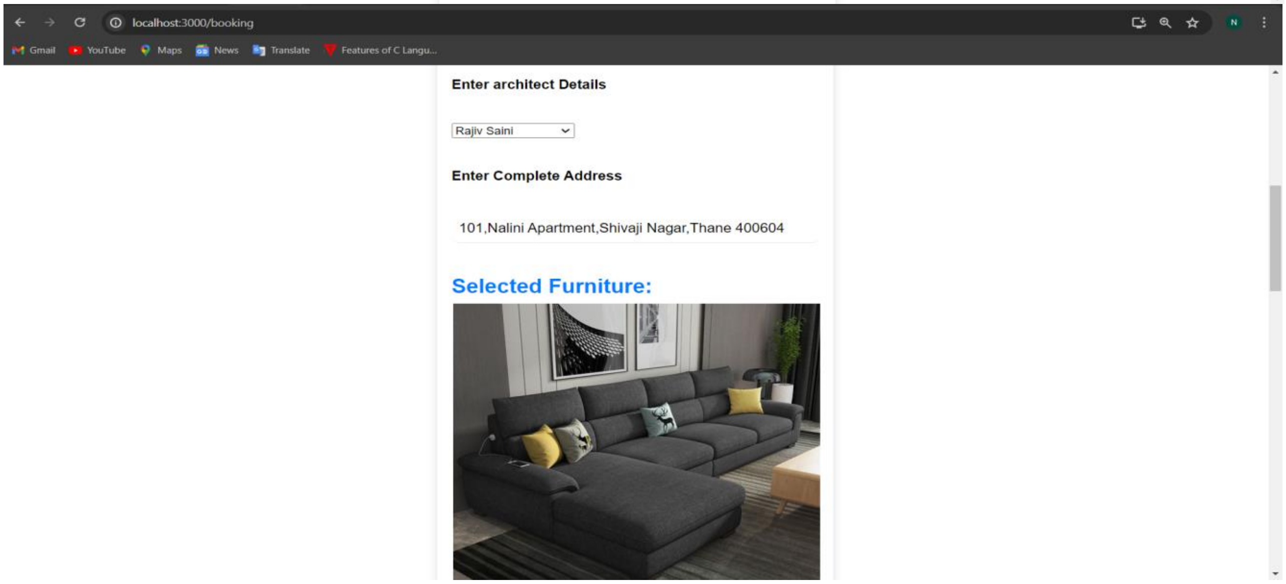
Sonu Shetty

Enter Your Contact Details

8454857103

Select A date

31-03-2024



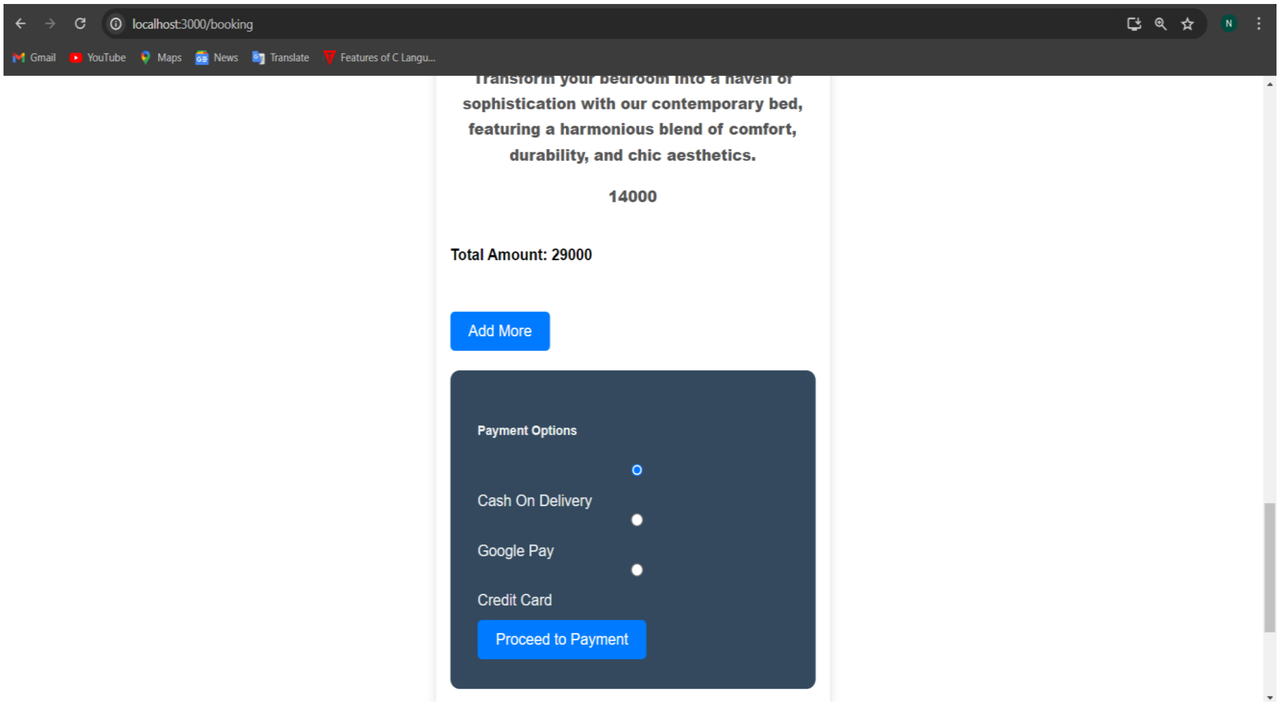
Enter architect Details

Rajiv Saini

Enter Complete Address

101,Nalini Apartment,Shivaji Nagar,Thane 400604

Selected Furniture:



Transform your bedroom into a haven of sophistication with our contemporary bed, featuring a harmonious blend of comfort, durability, and chic aesthetics.

14000

Total Amount: 29000

Add More

Payment Options

○ Cash On Delivery

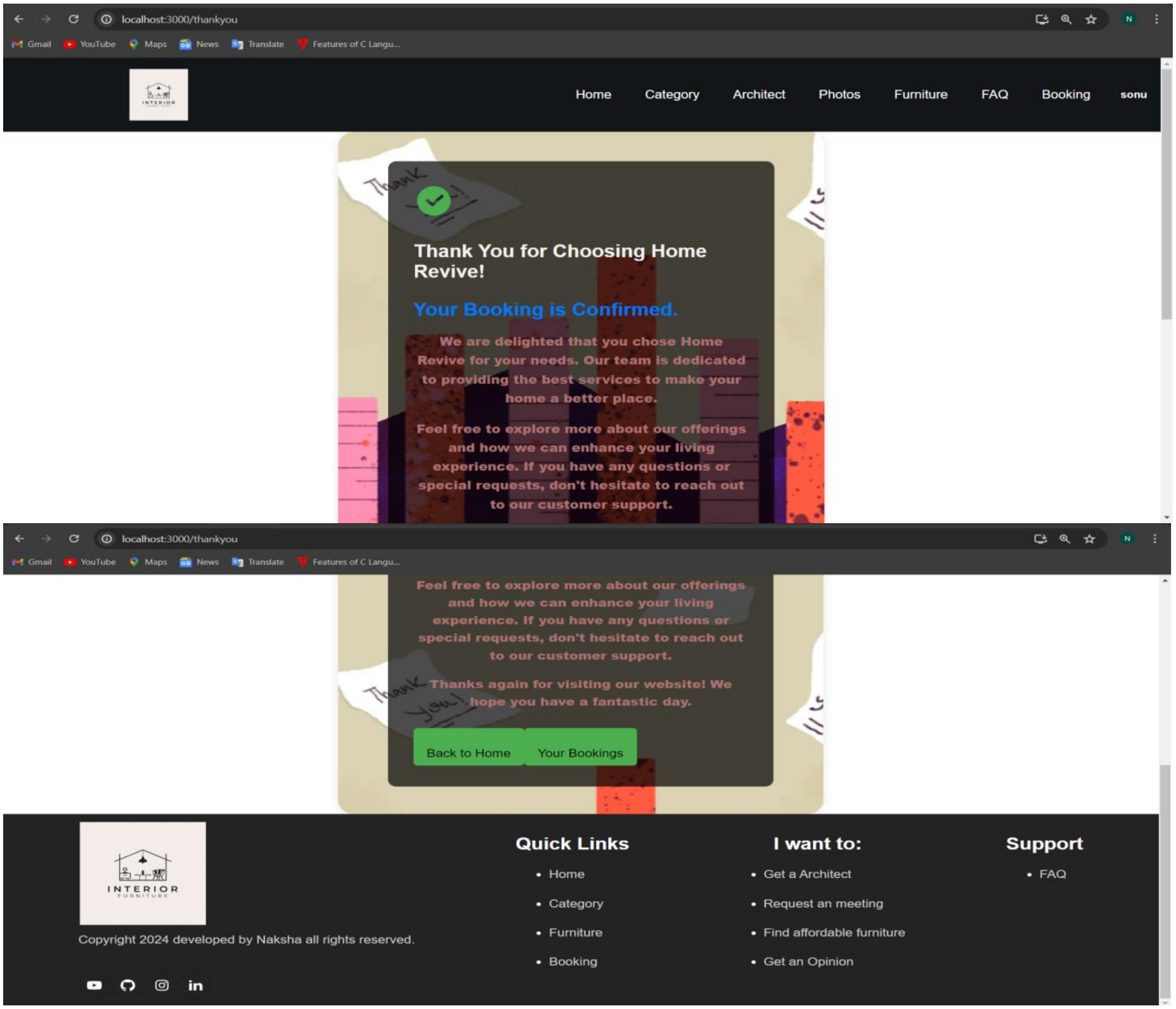
○ Google Pay

○ Credit Card

Proceed to Payment

6.2.8 Booking Page

Thank You page- The Thank You page on HomeRevive expresses gratitude to users for their engagement, providing a warm acknowledgment of their interaction.



Thank You for Choosing Home Revive!

Your Booking is Confirmed.

We are delighted that you chose Home Revive for your needs. Our team is dedicated to providing the best services to make your home a better place.

Feel free to explore more about our offerings and how we can enhance your living experience. If you have any questions or special requests, don't hesitate to reach out to our customer support.

Feel free to explore more about our offerings and how we can enhance your living experience. If you have any questions or special requests, don't hesitate to reach out to our customer support.

Thanks again for visiting our website! We hope you have a fantastic day.

Copyright 2024 developed by Naksha all rights reserved.

Quick Links

- Home
- Category
- Furniture
- Booking

I want to:

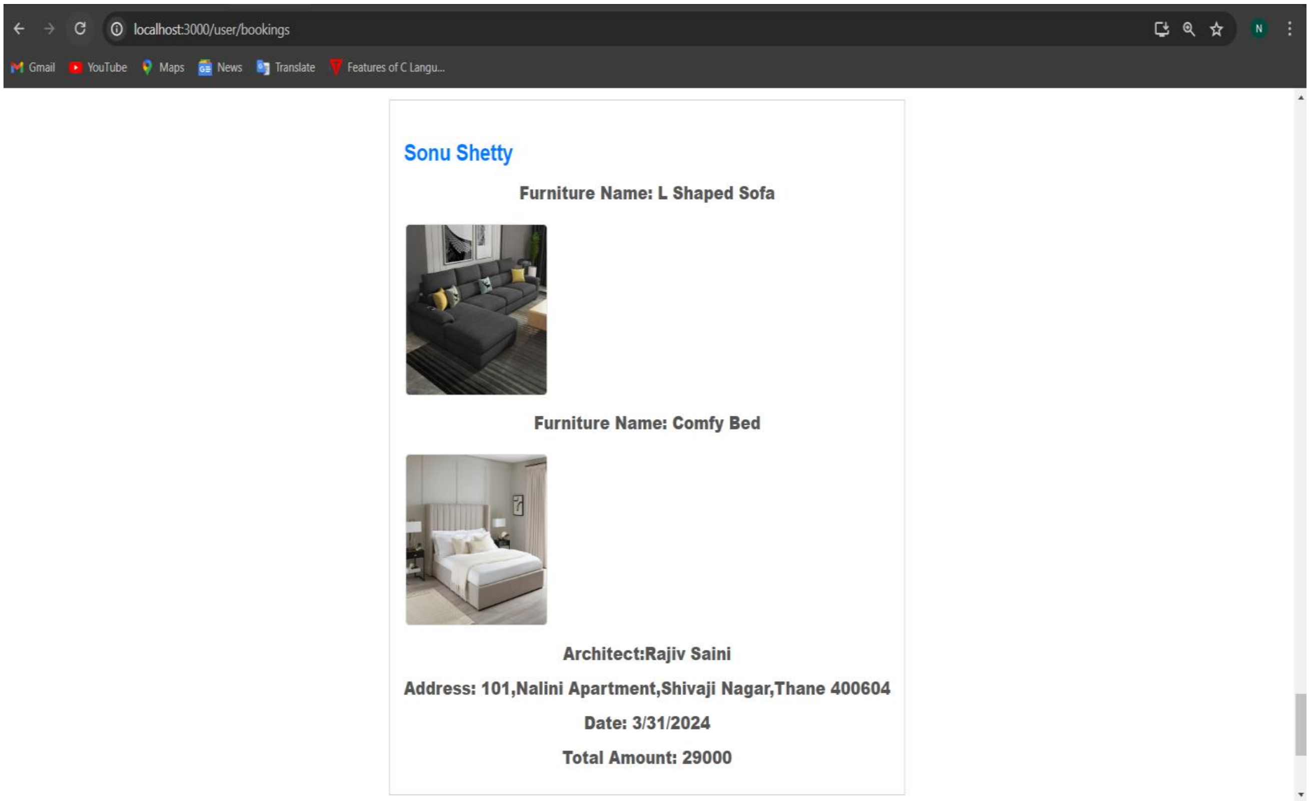
- Get a Architect
- Request an meeting
- Find affordable furniture
- Get an Opinion

Support

- FAQ

6.2.9 Thank You Page

Your Bookings- The "Your Bookings" section on HomeRevive provides users with a comprehensive overview of all their past and upcoming bookings, offering a convenient way to track and manage their renovation projects.



Sonu Shetty

Furniture Name: L Shaped Sofa

Furniture Name: Comfy Bed

Architect:Rajiv Saini

Address: 101,Nalini Apartment,Shivaji Nagar,Thane 400604

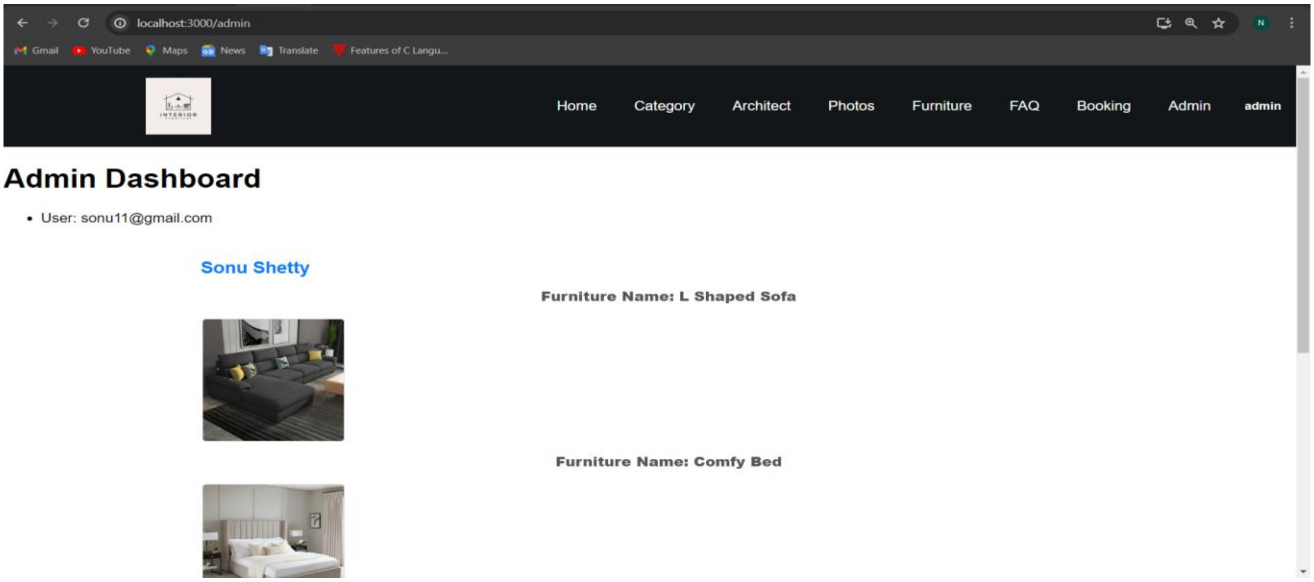
Date: 3/31/2024

Total Amount: 29000

6.2.10 Your Booking Page

Admin DashBoard - The Admin Dashboard page provides administrators with a simple overview of all user bookings, offering easy access to essential information for efficient management and tracking of reservations within HomeRevive.

[redacted]



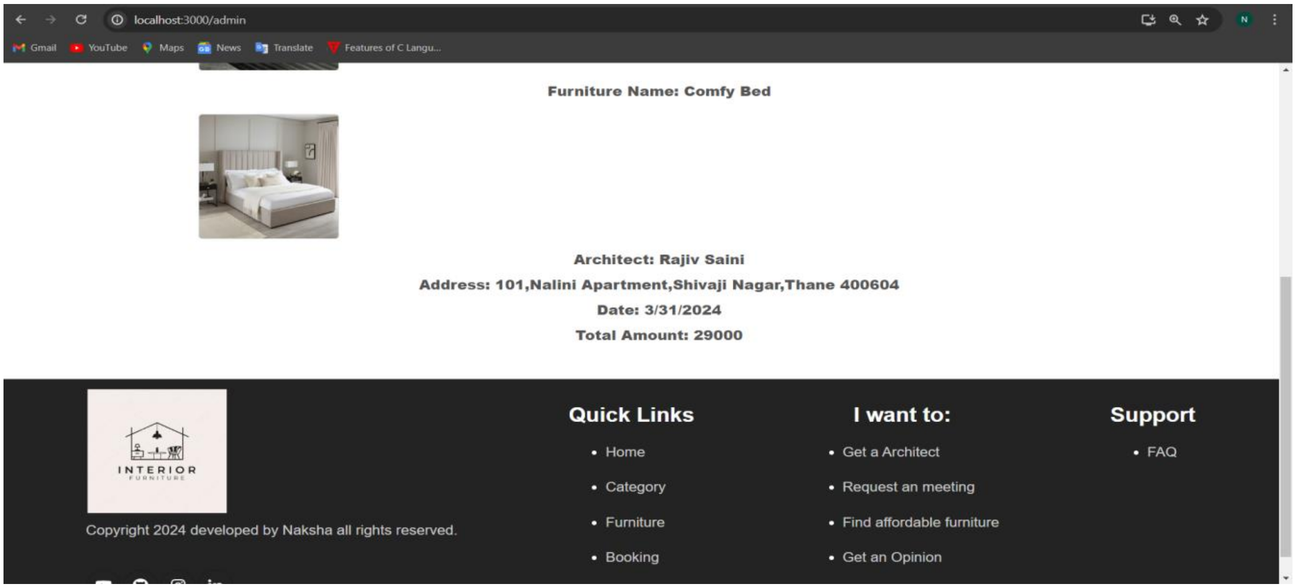
Admin Dashboard

- User: sonu11@gmail.com

Sonu Shetty

Furniture Name: L Shaped Sofa

Furniture Name: Comfy Bed



Furniture Name: Comfy Bed

Architect: Rajiv Saini

Address: 101,Nalini Apartment,Shivaji Nagar,Thane 400604

Date: 3/31/2024

Total Amount: 29000

Copyright 2024 developed by Naksha all rights reserved.

Quick Links

- Home
- Category
- Furniture
- Booking

I want to:

- Get a Architect
- Request an meeting
- Find affordable furniture
- Get an Opinion

Support

- FAQ

6.2.11 AdminDashBoard Page

Chapter 7: Conclusion

7.1. Significance of the system

The significance of the HomeRevive website lies in its holistic approach to fulfilling customers' dreams of creating their ideal home environment. By offering a comprehensive range of services and features, HomeRevive stands out as a one-stop destination for all aspects of home renovation and interior design.

At the heart of HomeRevive's mission is the goal of ensuring customer satisfaction. Through a combination of design inspiration, expert consultations, and seamless access to essential resources, the website aims to turn the vision of a dream home into a tangible reality. Whether customers seek modern aesthetics or a touch of vintage charm, HomeRevive caters to diverse tastes and preferences, acknowledging that each individual's idea of an ideal home is unique.

Central to HomeRevive's appeal is its emphasis on personalization. Customers have the opportunity to explore a wide range of design options across various categories such as living rooms, kitchens, bedrooms, and bathrooms. Additionally, they can engage in direct conversations with experienced contractors and architects to share their ideas and preferences, ensuring that every aspect of their home reflects their personal style and vision.

The convenience offered by HomeRevive cannot be overstated. By consolidating all necessary resources within a single platform, the website streamlines the home renovation process, saving customers valuable time and effort. From browsing design options to finding contractor details, accessing before-and-after pictures, and even purchasing furniture, HomeRevive simplifies every step of the journey towards creating the perfect home.

Moreover, HomeRevive is committed to enhancing the overall living experience by providing access to high-quality designs, skilled professionals, and luxurious furnishings. By empowering customers to transform their living spaces into havens of comfort and style, the website elevates the concept of home to new heights.

In essence, HomeRevive is more than just a website—it's a gateway to realizing the dream of a perfect home. Through its comprehensive services, dedication to customer satisfaction, emphasis on personalization, and commitment to convenience and luxury, HomeRevive redefines the home renovation experience, making dreams come true one project at a time.

7.2. Limitations of the system

Certainly, here's a condensed version of the limitations of the HomeRevive website:

Despite its many advantages, the HomeRevive website has certain limitations to consider. Firstly, while it offers options for personalization, the extent may be limited by predetermined design templates. Additionally, online communication may not fully replace face-to-face interaction, potentially affecting the depth of understanding between customers and professionals. Digital visualizations may not always accurately represent final outcomes, and the website's services may be restricted to specific geographic regions. Cost considerations and limited furniture selections may also pose challenges for some users. Finally, technical issues such as website downtime or compatibility problems could impact the user experience. Recognizing these limitations can help users manage expectations and make informed decisions when using HomeRevive.

7.3. Future scope of the project

Looking forward, the future scope of the HomeRevive project presents promising opportunities for growth and innovation. Firstly, expanding the geographical reach of HomeRevive's services could be a significant avenue for development. By establishing partnerships with contractors, architects, and suppliers in new regions, HomeRevive can extend its offerings to a broader customer base, catering to diverse needs and preferences across different locations.

Secondly, enhancing personalization options is crucial for the project's future success. Leveraging advanced design technologies and data analytics, HomeRevive can offer more tailored recommendations and customization features. This approach ensures that each customer's vision for their home is not only realized but surpassed, leading to increased satisfaction and loyalty.