A Mini Project Report

On

**BUILD AND CREW**

Submitted

*In partial fulfilment of the requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

**SK.NAYIMA KHATUN 22641A0591**

**TANGELLAPALLI SRAVYA 22641A0595**

**SEGGEM SWETHA 22641A0587**

**ALLA AJAY 22641A05A8**

Under the Guidance of

**Mr.Ch.Aravind kumar**

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI COLLEGE OF ENGINEERING

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Warangal – 506001

(2022-2025)

 VAAGDEVI COLLEGE OF ENGINEERING

(Autonomous)

(UGC Autonomous, Accredited by NAAC with “A”)

Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Accredited by NBA)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CERTIFICATE**

This is to certify that the project entitled **“BUILD AND CREW”** is submitted by

**Shaik Nayima khatun, Tangellapalli Sravya, Seggem Swetha,Alla Ajay bearing Hall Ticket Nos.22641A0591,22641A0595,22641A0587,22641A05A8** in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in Computer Science and Engineering during the academic year 2024-2025.

**GUIDE: HEAD OF THE DEPARTMENT:**

Mr.Ch.Aravind kumar Dr.N.Sathyavathi

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere gratitude and deep respect to **Dr.K.Prakash**, **Principal** of **Vaagdevi College of Engineering**, for providing us with all the necessary assistance and for his support and inspiration, which enabled us to carry out this project work at the institute.

Our heartfelt thanks extend to **Dr. N. Satyavathi, Associate Professor, Head of the Department,** Computer Science and Engineering,Vaagdevi College of Engineering, for equipping us with the required infrastructure and granting us the freedom to successfully conduct this project work.

We are immensely grateful to our guide**,** Mr.Ch.Aravind kumar**,** Assistant Professor,

Department of Computer Science and Engineering, for his/her constant support and invaluable guidance throughout the duration of this project work. Their insights and encouragement have been instrumental in our progress.

We would also like to thank the Project Coordinators, **Dr.K.Rekha Devi, Assistant Professor** and **Mr. Ch. Aravind Kumar, Assistant Professor**, for their thoughtful suggestions, motivation, and encouragement, which contributed significantly to the completion of our project.

Finally, we extend our sincere thanks to all the teaching and non-teaching staff members of the Department of Computer Science and Engineering, whose direct and indirect support was vital in ensuring the successful completion of this work.

**Sk.Nayima Khatun 22641A0591**

**T.Sravya 22641A0595**

**S.Swetha 22641A0587**

**A.Ajay 22641A05A8**

**DECLARATION**

We hereby certify that the work which is being presented in the project report entitled “**BUILD AND CREW”** in the partial fulfillment of the requirements for the award of the degree of **BATCHLOR OF TECHNOLOGY** and submitted in the **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, VAAGDEVI COLLEGE OF ENGINEERING**, is an authentic record of our own work carried out during the time period from 2022 to 2026 under the Guidance of **Mr.Ch.Aravind kumarv, Assistant Professor.** The matter presented in this project report has not been submitted by us for the award of any other degree of this or any other institute.

**Sk.Nayima Khatun 22641A0591**

**T.Sravya 22641A0595**

**S.Swetha 22641A0587**

**A.Ajay 22641A05A8**

## ABSTRACT

*The construction industry demands easy access to quality materials, skilled labor, and accurate project cost estimation. However, users often struggle with fragmented sources, price uncertainties, and time-consuming processes. This project proposes the development of a* ***Construction Marketplace Website****, designed to serve as an all-in-one platform where users can purchase a wide range of construction materials and hire professional workers such as plumbers, carpenters, electricians, painters, and more.*

*The website offers a user-friendly interface where customers can browse through categorized construction materials with detailed descriptions, pricing, and availability. Additionally, users can book workers based on their skill set, experience, and hourly rate, which is clearly displayed on the platform. To aid in project planning and budgeting, the website also provides an* ***estimation tool*** *that calculates the approximate cost of selected materials and the labor charges based on the project’s requirements.*

*By combining material procurement, labor booking, and cost estimation in a single digital platform, this project aims to streamline the construction process, improve transparency, and offer better control over budgets and timelines. It benefits individual homeowners, contractors, and builders by reducing effort, saving time, and ensuring access to reliable resources. The Construction Marketplace Website is thus envisioned as a modern, efficient solution to many traditional challenges in the construction industry.*

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGE NO.**

**CHAPTER-1: INTRODUCTION 1-3**

* 1. Project Objective 1
  2. Problem Statement 2

1.3 Digital Construction Marketplaces 2

1.4 Labor Hiring Platforms & Service Marketplaces 2

1.5 Cost Estimation Tools in Construction 3

1.6 Integrated Platforms &User-Centric Design 3

**CHAPTER-2: LITERATURE SURVEY 4-5**

2.1 Inefficiencies in Traditional Procurement 4

2.2 Specialized E-Commerce for Construction 4

2.3 Disconnection Between Labor Services and Material Supply 4

2.4 Challenges in Cost Estimation 4

2.5 Trust and Transparency in Construction Platforms 5

2.6 Technical Challenges in Platform Integration 5

2.7 Role of Mobile Technology in Construction Platforms 5

**CHAPTER-3: EXISTING SYSTEM 6-7**

**CHAPTER-4: PROPOSED SYSTEM 8-10**

4.1 Overview 8

4.2 System Architecture 9

**CHAPTER-5: SYSTEM REQUIREMENTS 11**

5.1 Functional Requirements 11

5.2 Non-functional Requirements 11

**CHAPTER-6: METHODOLOGY 12-14**

4.1 Requirement Analysis 12

4.2 System Design 12

4.3 Development Phase 13

4.4 Testing Phase 13

4.5 Deployment 14

4.6 Maintenance and Future Scope 14

**CHAPTER-07: MODULES DESCRIPTION 15-16**

5.1 Material Management Module 15

5.2 Worker Booking Module 15

5.3 Cost Estimation Module 15

5.4 User Interface Module 15

5.5 User Management Module 15

5.6 Communication and Support Module 16

5.7 Data Storage Module (JSON-Based) 16

**CHAPTER-08: DESIGNING 17-27**

8.1 Data Flow Diagram 17

* 1. 8.2 UML Diagrams 18-27

8.3.1 Class Diagram 19

8.3.2 Object Diagram 20

8.3.3 Sequence Diagram 21

8.3.4 Collaboration Diagram 22

8.3.5 Use Case Diagram 23

8.3.6 State Chart Diagram 24

8.3.7 Activity Diagram 25

8.3.8 Deployment Diagram 26

8.3.9 Component Diagram 27

**CHAPTER-9: SOURCE CODE 28-63**

**CHAPTER-10: IMPLEMENTATION 64-65**

10.1 Technology Stack 64

10.2 Material Procurement Implementation 64

10.3 Skilled Worker Booking Implementation 64

10.4 User Interface Development 65

10.6 JSON-Based Data Handling 65

10.6User Management Implementation 65

10.7 Integration and Testing 65

**CHAPTER-11: RESULTS 66-67**

**CHAPTER-12: TESTING 68**

**CHAPTER-13: CONCLUSION AND FUTURE SCOPE 69**

13.1 conclusion

13.2 Future Scope

**CHAPTER-14: REFERENCES 70**

**LIST OF FIGURES**

**FIGURE NAMES PAGE NO**

Fig.8.1 Data Flow Diagram 17

Fig. 8.3.1 Class Diagram 19

Fig. 8.3.2 Object Diagram 20

Fig. 8.3.3 Sequence Diagram 21

Fig. 8.3.4 Collaboration Diagram 22

Fig. 8.3.5 Use Case Diagram 23

Fig. 8.3.6 State Chart Diagram 24

Fig. 8.3.7 Activity Diagram 25

Fig. 8.3.8 Deployment Diagram 26

Fig. 8.3.9 Component Diagram 27

Fig.11.1 Login Page 66

Fig.11.2 Home Page 66

Fig.11.3 Main Page 67

## CHAPTER-1

**INTRODUCTION**

This chapter introduces the concept of a centralized construction marketplace website by exploring existing digital solutions related to material procurement, labor booking, and cost estimation in the construction domain. It highlights how current tools, while effective in specific areas, often operate in silos, lacking an integrated experience for end-users. This chapter demonstrates how the proposed platform addresses these limitations by providing a unified digital interface for accessing materials, hiring skilled workers, and calculating estimated costs.

### 1.1 Project Objective

The objective of the Construction Marketplace Website project is to develop an integrated, user-friendly platform that simplifies and streamlines the process of purchasing construction materials, hiring skilled workers, and estimating project costs. The platform aims to offer a wide range of construction materials with detailed information on pricing and availability, making it easy for users to compare and purchase products. It also enables users to book skilled workers, such as plumbers, carpenters, electricians, and painters, by providing access to worker profiles, experience levels, ratings, and hourly charges. A key feature of the platform is the real-time cost estimation tool, which helps users plan and manage their budgets effectively by calculating the approximate costs of materials and labor.

In addition to serving individual users, the website also supports material dealers by offering them a digital space to showcase their products and reach a broader customer base, helping to boost their business. Similarly, skilled workers benefit by finding more consistent job opportunities through direct connection with customers. By ensuring transparency in pricing and offering a simple, intuitive interface, the project aims to improve trust and convenience for all users. Ultimately, the Construction Marketplace Website seeks to modernize the construction industry’s sourcing and hiring practices, creating a more efficient, transparent, and accessible environment for homeowners, contractors, suppliers, and workers alike.

### 1.2 Problem Statement

The construction industry has long faced significant challenges in sourcing materials and hiring skilled labor, often resulting in delays, inefficiencies, and budget overruns. Homeowners and contractors are frequently required to visit multiple suppliers to purchase construction materials, leading to time-consuming searches and inconsistent pricing. Additionally, hiring qualified workers for specialized tasks such as plumbing, carpentry, and electrical work remains a fragmented process, with a lack of reliable platforms for users to compare worker profiles, experience, and rates. Furthermore, the inability to easily estimate the total cost of materials and labor complicates project budgeting and financial planning, causing uncertainty and cost discrepancies.

This project addresses these issues by providing a comprehensive online platform that integrates material purchasing, labor booking, and cost estimation in one place. By offering real-time price transparency for materials and labor, along with the ability to hire trusted professionals, the Construction Marketplace Website seeks to simplify and streamline the construction process, reduce inefficiencies, and enable more accurate project planning and budgeting

**1.3 Digital Construction Marketplaces**

In recent years, the digitalization of the construction sector has led to the development of platforms like BuildMart, Infra Bazaar, and MaterialTree, which focus primarily on the online procurement of construction materials. These platforms offer benefits such as product filtering, delivery tracking, and digital payments. However, their scope remains limited to e-commerce for materials, without offering integrated labor services or cost planning features. The proposed Construction Marketplace Website expands this concept by including both materials and workforce booking in a single system, thereby simplifying the overall planning and execution process for users.

**1.4 Labor Hiring Platforms and Service Marketplaces**

Service-based platforms like UrbanClap (now Urban Company), TaskRabbit, and HouseJoy have made it easier for users to find and book professionals for home services such as plumbing, carpentry, and electrical work. While these platforms support labor hiring, they typically do not cater specifically to the construction industry, nor do they allow the bundling of labor with materials or cost estimates. The proposed platform is unique in combining material selection with location-based labor hiring, offering users an end-to-end solution for construction and renovation needs.

**1.5 Cost Estimation Tools in Construction**

Accurate cost estimation is essential for construction projects of any scale. Tools such as Building Information Modeling (BIM), cost calculators, and project management software are often used by professionals to forecast expenses. However, such tools are generally technical and not accessible to average homeowners or small-scale users. The Construction Marketplace Website bridges this gap by offering a user-friendly estimation module that automatically calculates the expected cost of materials and labor based on user inputs like area, quantity, and work duration, making financial planning simpler and more accurate for non-technical users inputs like area, quantity, and work duration, making financial planning simpler and more accurate for non-technical users.

**1.6 Integrated Platforms and User-Centric Design**

While some platforms offer standalone features—material procurement, labor services, or estimation tools—there is a lack of integration and personalization across these systems. Research in digital platform design emphasizes the importance of user-centric architecture that combines functionality, ease of use, and efficiency (Norman, 2013). The Construction Marketplace Website is designed to be intuitive, mobile-friendly, and inclusive of all key services in one place, reducing fragmentation and improving the user experience for clients and workers alike.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 Inefficiencies in Traditional Procurement**

Traditional methods of acquiring construction materials often involve visiting multiple physical suppliers, leading to time-consuming and inefficient processes. While general e-commerce platforms such as Amazon and Lowe’s offer some construction materials, they lack integration with services like labor hiring, which limits their applicability in holistic construction management.

**2.2 Specialized E-Commerce for Construction**

Davis et al. (2017) argue that dedicated e-commerce platforms tailored specifically for construction materials significantly enhance procurement efficiency and contribute to better project timelines, as they offer features like bulk ordering, supplier transparency, and inventory filtering specific to industry needs.

**2.3 Disconnection Between Labor Services and Material Supply**

Popular service platforms such as TaskRabbit and UrbanClap provide easy access to skilled labor, including plumbers, carpenters, and electricians, yet they do not incorporate the procurement of materials within their systems. Zhao et al. (2018) recommend that integrating both material supply and labor booking on a unified platform would greatly streamline the construction and renovation process.

**2.4 Challenges in Cost Estimation**

Accurate cost forecasting continues to be a critical issue in the construction domain due to market-driven price fluctuations. Brown and Yates (2015) identify the lack of reliable tools as a key factor leading to budget overruns. Wang et al. (2020) support the integration of real-time pricing data into cost estimation tools, highlighting its impact on improved budget planning and reduced financial risk.

**2.5 Trust and Transparency in Construction Platforms**

User trust plays a vital role in the adoption of digital platforms for construction services. Li et al. (2019) stress that features such as user reviews, service ratings, and transparent pricing models are crucial in building confidence among both consumers and service providers, leading to greater platform reliability and long-term engagement.

**2.6 Technical Challenges in Platform Integration**

Combining diverse services like material procurement, labor booking, and dynamic cost estimation into one cohesive system poses several technical difficulties. Chen et al. (2020) note that managing real-time data updates, ensuring secure digital payments, and maintaining system performance under load are key technical hurdles that must be addressed during platform development.

**2.7 Role of Mobile Technology in Construction Platforms**

With the rise of mobile-first applications, the construction industry has benefited from improved communication, quicker decision-making, and on-site data access. Miller and Jackson (2018) highlight how mobile platforms support real-time updates and collaboration, reinforcing the need for responsive, mobile-optimized systems in construction-related digital solutions.

**CHAPTER-3**

### EXISTING SYSTEM

In the current digital ecosystem, various platforms cater to specific aspects of the construction industry, such as material procurement, skilled labor booking, or project cost estimation; however, a fully unified solution that integrates all three components is rare. E-commerce giants like **Amazon** and **Flipkart** offer a wide range of construction-related materials, including tools, fixtures, and basic hardware items, but these platforms lack features related to construction labor services or real-time project cost estimation. Similarly, hardware-focused platforms such as **BuildNext** or **Materialtree** specialize in construction material supply, providing users with access to a wide range of raw materials including cement, sand, aggregates, and bricks. These platforms also offer location-based delivery services and allow users to compare prices across brands and sellers. However, these services do not extend into labor hiring or dynamic budget estimation based on changing prices and work requirements.

On the other hand, platforms like **Urban Company (formerly UrbanClap)** and **Housejoy** primarily focus on skilled labor services, offering the ability to book professionals for tasks such as plumbing, carpentry, electrical work, painting, and home cleaning. These platforms emphasize user reviews, verified service providers, transparent pricing, and scheduling flexibility. However, they operate solely in the services domain and do not allow users to purchase building materials or get project-wide cost estimates based on both material and labor inputs. Moreover, while Urban Company provides an hourly rate model and transparent service pricing, it lacks a comprehensive tool for calculating total construction or renovation costs involving both services and materials.

Some regional platforms like **InstaOffice** or **Brick&Bolt** attempt to bridge this gap by offering end-to-end construction solutions. Brick&Bolt, for example, provides home construction and renovation packages with estimated costs, timelines, and material details. However, these systems are typically contract-based and do not offer the modular flexibility for users to purchase materials separately or book individual labor services based on specific needs. They also often require customers to commit to full-scale projects rather than enabling DIY or small-scale renovations that require flexible budgeting.

Additionally, none of the mainstream platforms provide a real-time **cost estimation tool** that dynamically updates based on changes in material prices and labor rates. This lack of integration leads to inefficiencies, forcing users to switch between multiple platforms to estimate costs, procure materials, and hire labor. As highlighted in various studies, the fragmented nature of these services results in increased project timelines, inflated costs, and limited transparency for end-users.

Therefore, while existing systems offer strong features in isolation—either in material procurement or skilled labor booking—they fall short of providing an integrated, scalable, and user-centric construction marketplace. This gap presents a valuable opportunity for a platform that unifies material listing, labor booking, and real-time cost estimation into a single streamlined website, allowing users to plan, source, and execute construction projects of any scale with increased control, transparency, and efficiency.

**Limitations of Existing System**

1. **Fragmentation:** Users must visit multiple platforms for materials and labor, causing inefficiencies and delays.
2. **Lack of Integration:** No unified platform combining materials, labor services, and cost estimation.
3. **Limited Pricing Transparency**: Real-time pricing for materials and labor is often unavailable or unclear.
4. **No Cost Estimation:** Existing systems do not offer integrated cost estimation for both materials and labor.
5. **Limited Skilled Labor Availability:** Platforms for labor booking often lack construction-specific workers or quality verification.

### CHAPTER-4

**PROPOSED SYSTEM**

**4.1 Overview**

The proposed system is a comprehensive and user-centric web-based **Construction Marketplace Website** designed to streamline the traditionally fragmented process of managing construction projects. This platform serves as a one-stop digital solution that combines three essential components: **construction material procurement**, **skilled labor booking**, and **dynamic project cost estimation**. The system is aimed at homeowners, contractors, and project managers who often face inefficiencies in locating trustworthy material suppliers, hiring verified workers, and estimating budgets. By integrating these core functionalities into a single platform with a responsive user interface, the system simplifies project planning and execution while saving time, reducing costs, and increasing transparency.

Instead of relying on complex databases, the platform uses lightweight and structured **JSON files** to store and manage data related to materials, workers, pricing, availability, and bookings. This enhances performance and simplifies the backend architecture, making it more accessible for future development and scalability. Users can browse and purchase a wide range of construction materials, search for skilled workers such as plumbers and carpenters, and get a real-time estimate of their project costs based on up-to-date prices and labor charges. The website also incorporates secure payment options and a built-in communication module to ensure smooth transactions and coordination between parties. Overall, the proposed system creates a unified digital environment that supports every stage of the construction process, from planning to procurement to labor execution.

**4.2 System Architecture**

The proposed Construction Marketplace Website is built on a client-server architecture. This model separates the user-facing interface from the backend services, ensuring efficient processing, scalability, and a user-friendly experience. The architecture includes clearly defined modules on both the client and server sides, supported by a secure and structured database and integrated with external services.

**1. Client-Side Architecture**

The client-side represents the front-end layer of the system, where users interact with the application through a web or mobile interface. This layer is designed to be intuitive and responsive, offering easy navigation for different user roles including customers, suppliers, and skilled workers.

Key features on the client side include the Login and Registration module, which allows users to create accounts and access role-based functionalities. The Material Procurement Interface enables users to browse, search, and compare a wide variety of construction materials such as cement, sand, bricks, plumbing equipment, and electrical items. Real-time pricing and availability are displayed for better decision-making.

The Labor Booking Interface allows customers to find skilled professionals like plumbers, carpenters, and electricians. Worker profiles display ratings, hourly rates, experience, and certification details. Users can filter professionals based on location, specialization, and availability.

A well-organized dealer and worker Dashboard presents order histories, bookings, Inventory.

**2. Server-Side Architecture**

The server-side handles all core functionalities, processing requests, managing data, and ensuring secure communication between the front end and the database. It also houses the system's business logic, enabling dynamic operations and automated workflows.

The Authentication and User Management Module verifies user credentials, manages user roles, and ensures secure access control. The Material Management Module oversees the storage and retrieval of material data, including price updates, supplier information, and inventory tracking.

The workers Booking Module manages skilled worker profiles, availability schedules, ratings, and booking transactions. The Cost Estimation Engine processes user inputs and fetches live data from the material and labor modules to deliver updated project cost estimates.

**3. Data Storage in JSON Files**

All data, including user profiles, materials, workers are stored in JSON files. Each type of data is stored in its own JSON file, with clear and structured fields to enable quick access and efficient updates.

**User Data (JSON File):** Stores details such as usernames, roles (customer, supplier, worker), login credentials, and profile information.

**Material Data (JSON File):** Contains information about materials, including name, price, availability, and supplier information.

**Workers Data (JSON File):** Stores worker profiles, including hourly rates, ratings, skills, and location.

.

1. Top of Form
2. Bottom of Form

### CHAPTER-5

### SYSTEM REQUIREMENTS

### Functional Requirements

* **User Registration & Login:** Users can register, log in, and manage their profiles.
* **Material Listing & Purchase:** Browse, compare, and order construction materials with real-time price and availability.
* **Worker Booking:** Book skilled labor (plumber, carpenter, etc.) based on availability, location, and hourly rate.
* **Cost Estimation:** Automatically calculate total cost for selected materials and labor.
* **Messaging & Support:** Users can communicate with suppliers or workers and get customer support.
* **JSON-Based Storage:** All data is stored and managed using JSON files instead of a traditional database.

### Non-functional Requirements

* **Performance:** Fast response and smooth user experience.
* **Usability:** Easy-to-use and mobile-friendly interface.
* **Scalability:** Easily extendable to more materials, workers, and regions.
* **Security:** Secure login and encrypted payment handling.
* **Reliability:** Stable operation and error handling.
* **Maintainability:** Simple structure using JSON for easy updates.
* **Availability:** High uptime with minimal downtime.
* **Data Integrity:** Valid and consistent data across the system.

**CHAPTER-6**

**METHODOLOGY**

The methodology adopted for the development of the Construction Marketplace Website is a structured, step-by-step approach designed to ensure that the platform effectively meets the requirements of users looking to procure construction materials, book skilled labor, and receive accurate cost estimations. The development process follows a modified Waterfall Model, incorporating both planning and adaptability to meet evolving user expectations.

**6.1 Requirement Analysis**

* Identified core features based on real-world needs of homeowners, contractors, and construction service seekers.
* Studied user behaviour for purchasing materials and booking labor such as plumbers, electricians, and carpenters.
* Requirements were categorized into:
  + Material procurement
  + Skilled worker booking
  + Estimation system
  + User communication and support
  + Data storage using JSON

**6.2 System Design**

* **Architecture Planning:** Designed with a two-tier architecture (Client-Side and Server-Side).
* **Client-Side Design:**
  + Developed UI using HTML5, CSS3, and JavaScript.
  + Ensured responsiveness and usability across devices.
  + Designed user-friendly forms for material selection and worker booking.
* **Server-Side Design:**
  + Used JSON files for data storage (materials, workers, bookings).
  + Scripts written in JavaScript to read, write, and update JSON data.
  + Emphasis on data structure and access efficiency.

**6.3 Development Phase**

* **Material Module:**
  + Developed a catalog of construction materials including cement, bricks, pipes, wires, etc.
  + Integrated real-time search and filter options.
  + Pricing and availability fetched from structured JSON files.
* **Worker Booking Module:**
  + Implemented worker profiles with details: skill type, experience, hourly charge, availability.
  + Enabled users to search, view ratings, and book workers.
* **Cost Estimation Tool:**
  + Auto-calculates total cost based on selected materials.
  + Dynamic updating of estimate as user modifies selections.
* **JSON File Management:**
  + Created reusable, well-structured JSON files.
  + Implemented efficient read/write operations with error handling.

**6.4 Testing Phase**

* **Unit Testing:** Tested individual modules such as the material catalog, booking system, and estimation logic independently to ensure that each component functioned correctly and met the design requirements.
* **Integration Testing:** Verified the interaction and data flow between modules, ensuring a seamless user experience from selecting materials and workers to generating cost estimations and confirming bookings.
* **UI Testing:** Ensured that the user interface was responsive, visually consistent, and easy to navigate across different devices and screen sizes, providing a smooth and intuitive user experience.
* **Validation Testing:** Checked all user inputs and form submissions to prevent missing or invalid data, ensuring that only correctly formatted and complete information could be processed by the system.

**6.5 Deployment**

* Deployed the system on a local server or online hosting environment for demonstration.
* JSON files uploaded as flat data storage and accessed by backend logic.
* Verified performance, availability, and integrity in real-time access scenarios.

**6.6** **Maintenance and Future Scope**

* System designed for scalability to add:
  + More categories of materials and workers
  + Location-based filtering
  + Real-time chat and reviews
* JSON file structure makes it easy to edit data without needing database migration**.**

**CHAPTER-7**

**MODULES DESCRIPTION**

**7.1 Material Management Module**

* Catalogs and allows searching, filtering, and purchasing materials.
* Real-time availability and pricing, stored in JSON files.
* Includes product details and images.

**7.2 Worker Booking Module**

* Lists workers by profession with detailed profiles.
* Allows booking based on trade, location, and availability.
* Real-time updates on worker status and bookings.

**7.3 Cost Estimation Module**

* Calculates project costs based on materials and worker hours.
* Provides itemized breakdown and updates dynamically.

**7.4 User Interface Module**

* Responsive, intuitive design using HTML5, CSS3, and JavaScript.
* Segregated pages for materials, workers, estimates, and profiles.

**7.5 User Management Module**

* Handles registration, login, and profile management.
* Tracks past bookings, order history, and saved estimates.

**7.6 Communication and Support Module**

* Enables messaging between users, suppliers, and workers.
* Includes customer support and issue resolution.

**7.7 Data Storage Module (JSON-Based)**

* Uses JSON files to store materials, workers, user data, and bookings.
* Efficient data access with no need for a database server.

**CHAPTER-8**

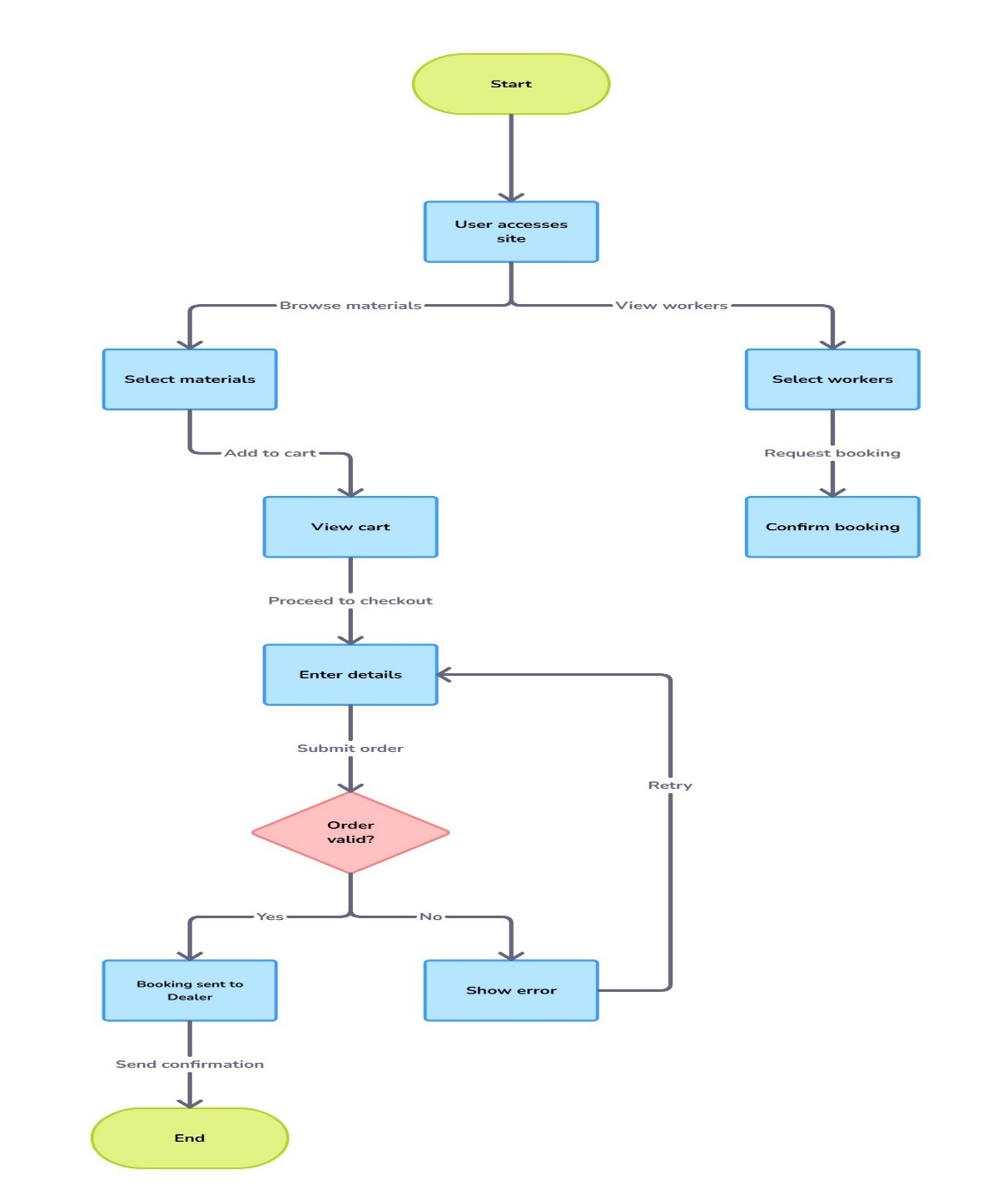
**SYSTEM DESIGN**

**8.1 Data Flow Diagram**

Website interaction flow for:

* Materials: Browse → Cart → Checkout → Details → Order (validated) → Booking/Confirmation.
* Workers: View → Select → Book → Confirm.

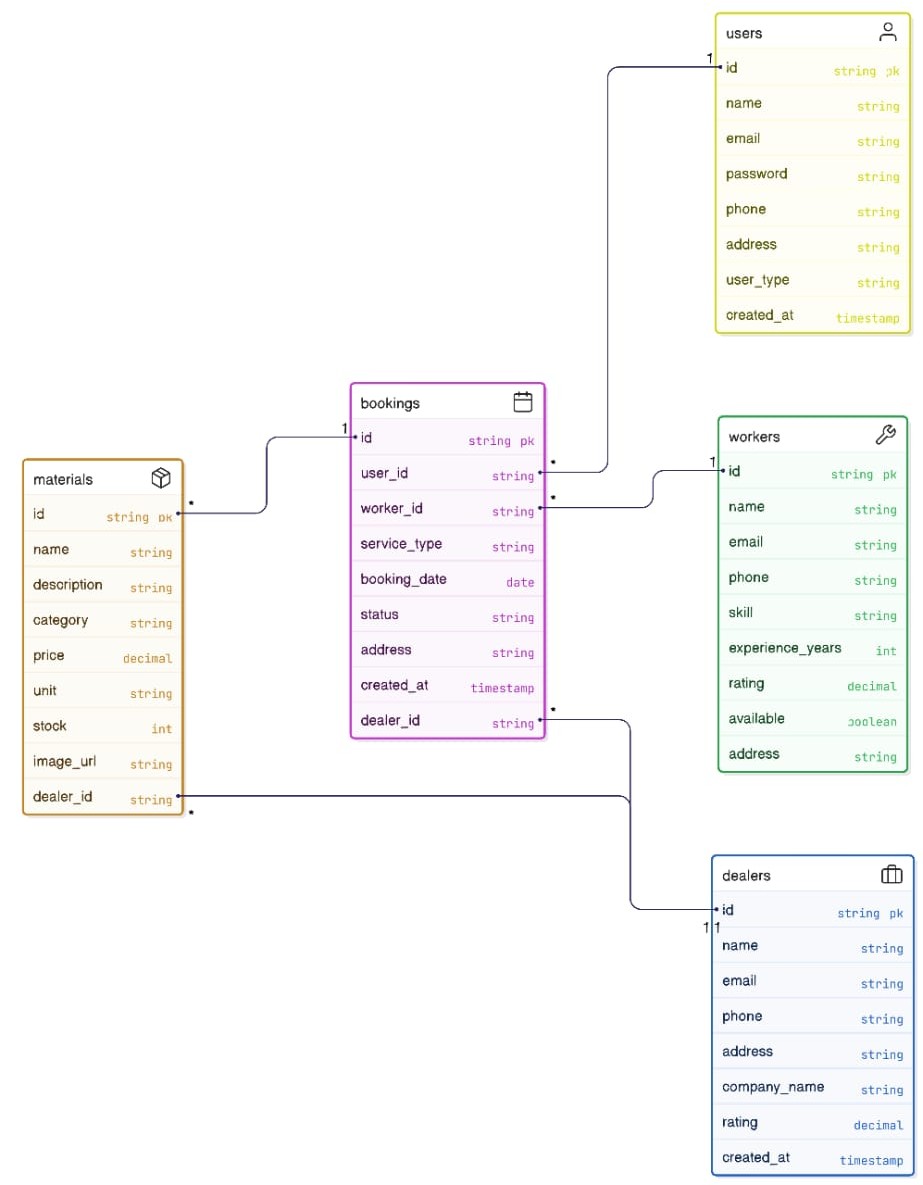
Both processes conclude at the end.



**Fig- 8.1 Data Flow Diagram**

**8.2 ER Diagram**

This ER diagram represents the data model of a “Construction Marketplace” system, where the “users” table stores general information about all system users such as customers, workers, dealers, and admins, distinguished by the user\_type attribute; the “workers” table holds specific details related to laborers like skills, experience, availability, and ratings, and is linked to “bookings” through worker\_id, allowing customers to hire them for specific construction tasks; the “dealers” table maintains records of material suppliers, including company details and contact information, and is associated with both materials—which contains comprehensive data on construction items such as name, category, price, stock, and dealer ID—and bookings, indicating which dealer supplied the materials or services for a given transaction; lastly, the “bookings” table acts as a central link connecting users, workers, and dealers by recording each service request’s details such as service type, date, status, and address, forming the core transactional entity in the marketplace.



**Fig-8.2 ER Diagram**

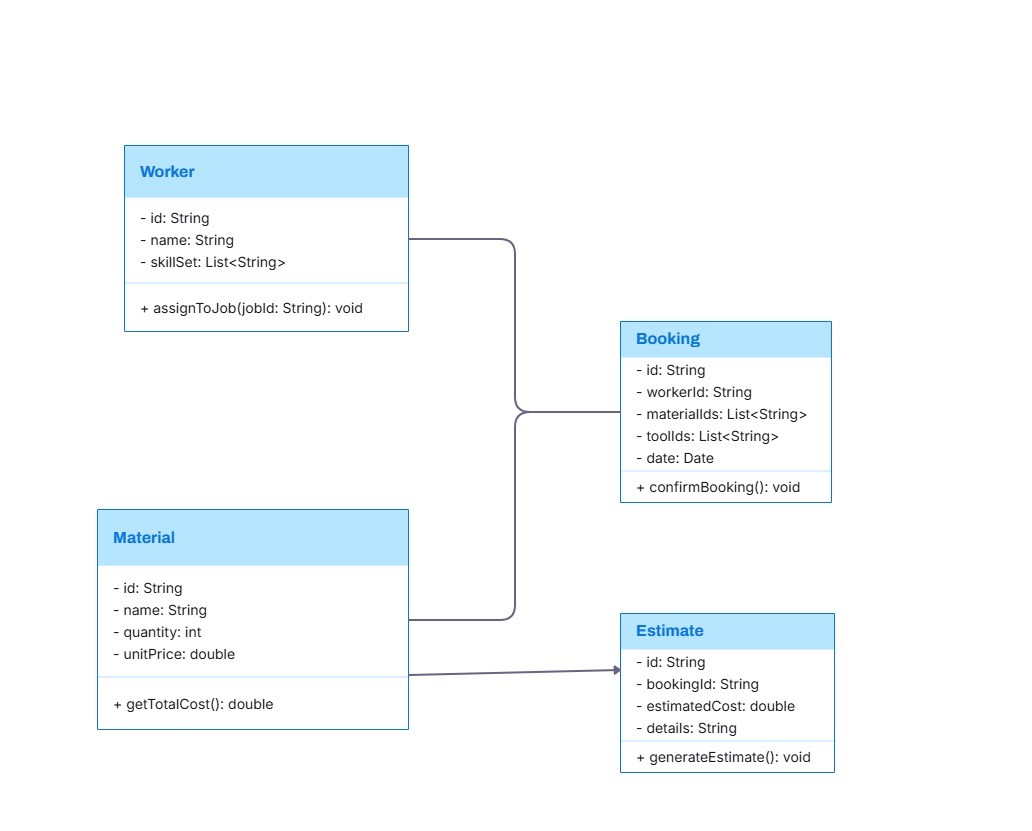
**8.3 UML DIAGRAMS**

Unified Modeling Language (UML) is a standardized visual language used to model, design, and document the architecture of software systems. It provides a set of graphical notations to represent the structure and behavior of a system, making complex systems easier to understand and communicate among developers, stakeholders, and business analysts.

**Key Points About UML**

* **Standardized Notation:** UML offers a universal set of symbols and diagrams that standardize how software systems are described, which helps teams speak the same language regardless of their background or the programming language they use.
* **Types of Diagrams:** UML includes various diagrams that can be categorized into two main types:
* **Structural Diagrams**: These describe the static aspects of the system. Examples include Class Diagrams, Component Diagrams, and Deployment Diagrams.
* **Behavioral Diagrams**: These focus on the dynamic aspects and interactions within the system. Examples include Use Case Diagrams, Sequence Diagrams, Activity Diagrams, and Collaboration Diagrams.
* **System Documentation and Communication:** UML serves as an effective tool for documenting system requirements, design decisions, and the overall architecture. It helps bridge the gap between technical and non-technical stakeholders by providing clear, visual representations of the system.
* **Design and Analysis:** By modeling different aspects of a system, UML enables developers to analyze and validate the design early in the development process. This can lead to better decision-making, reduced complexity, and improved system maintainability.
* **Flexibility:** UML is versatile and can be used across a wide range of applications, from small-scale projects to large, complex systems. It supports object-oriented design principles and can be adapted to various methodologies such as Agile or Waterfall.

**8.3.1 Class Diagram**

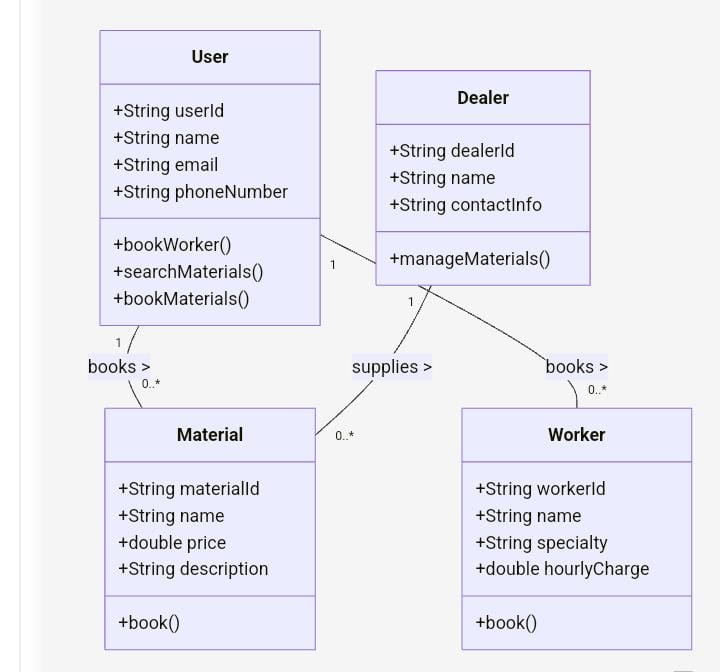
This class diagram illustrates the relationships and attributes of four classes in a system: Worker, which manages workers and their skills; Booking, which links workers to jobs with associated materials and dates; Material, detailing the characteristics of materials including quantity and price; and Estimate, which generates cost estimates for bookings. Each class includes specific properties and methods to facilitate functionalities like job assignment, booking confirmation, material cost calculation, and estimate generation. 

**Fig-8.3.1 Class Diagram**

**8.3.2 Object Diagram**

This diagram shows four main categories: Users, Dealers, Materials (like books), and Workers. Each category has some information associated with it, like a User having an ID, name, email, and phone number. It also shows what actions each category can perform, for example, a User can "book Materials". The lines connecting these categories show how they relate to each other. For instance, a single User can book many different Materials, and a single Material can be booked by one User.

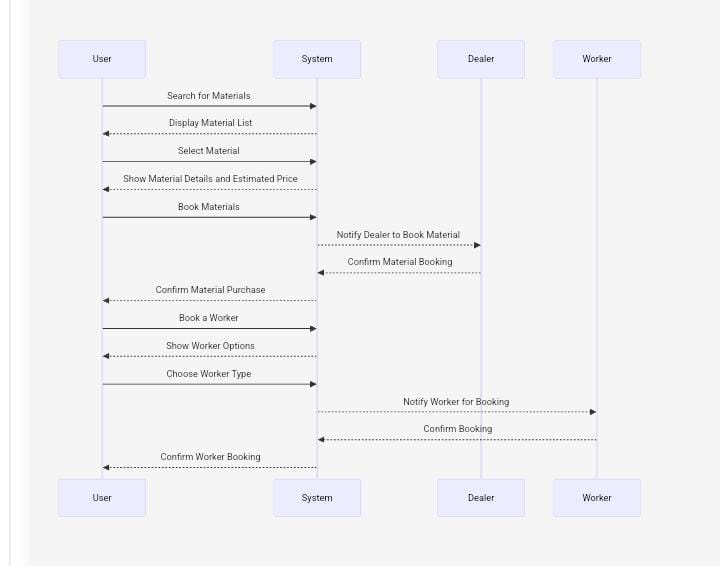
Similarly, a single Dealer provides many Materials, and a single Material is provided by one Dealer. Also, a single Dealer can book many Workers, and a single Worker can be booked by one Dealer. So, this diagram gives us a basic understanding of the different roles and items in this system and how they are linked.



**Fig-8.3.2 Object Diagram**

**8.3.3 Sequence Diagram**

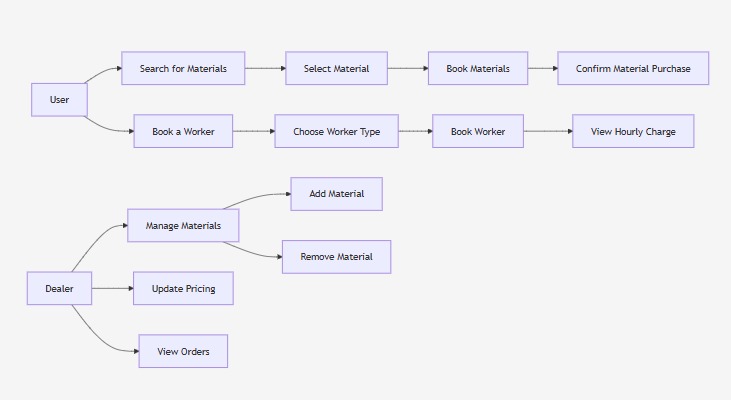
This sequence diagram meticulously delineates the sequential interactions commencing with a User searching for materials within a System, leading to the display of a material list, the User's selection, the presentation of detailed information and an estimated price, culminating in the User booking the materials, which triggers a notification to the Dealer for booking confirmation, subsequently relayed back to the User as a purchase confirmation; following this material procurement, the User initiates a separate process to book a worker, prompting the System to present worker options, enabling the User to choose a worker type, which then results in the System notifying the Worker for booking, receiving the Worker's confirmation, and finally informing the User of the confirmed worker booking, thereby illustrating a comprehensive workflow for both material acquisition and labor arrangement through the intermediary System.



**Fig-8.3.3 Sequence Diagram**

**8.3.4 Collaboration Diagram**

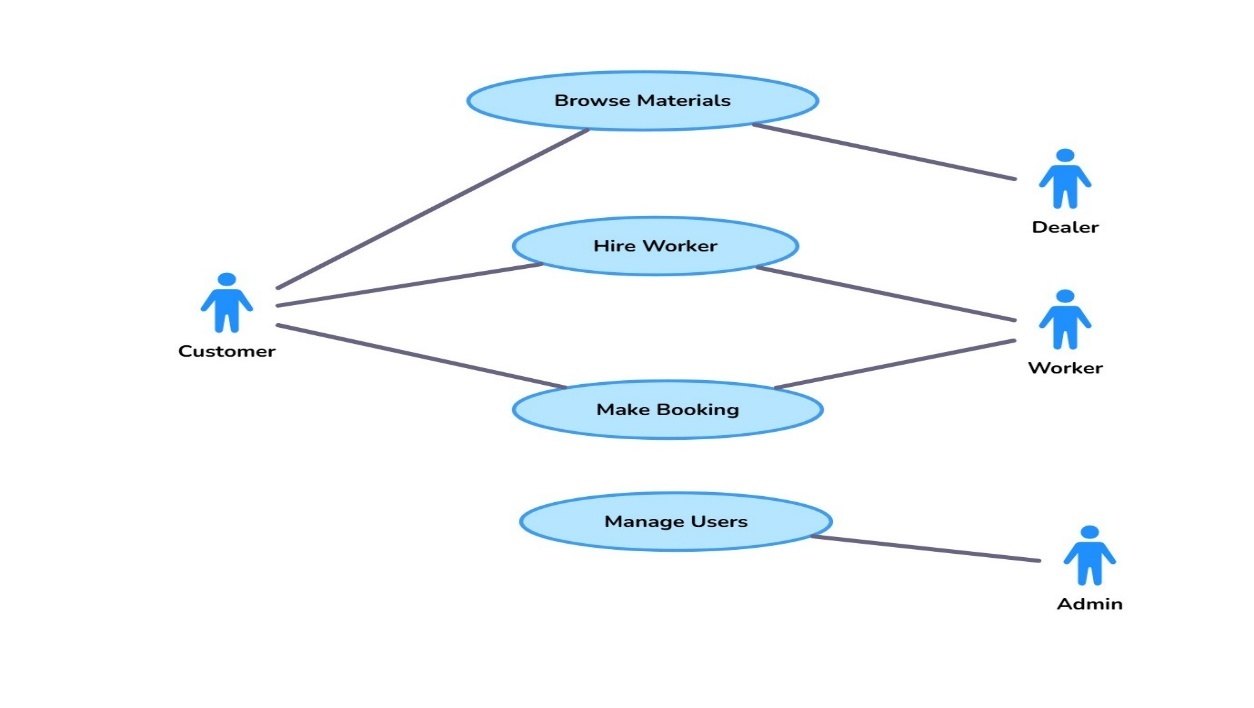
The diagram illustrates that a User can either search for materials, then select a material, proceed to book the materials, and finally confirm the material purchase, or alternatively, the User can book a worker, then choose the worker type, subsequently book the worker, and then view the hourly charge. On the other hand, the Dealer can manage materials, which includes the ability to add material and remove material, as well as update pricing and view orders.



**Fig-8.3.4 Collaboration Diagram**

**8.3.5 Use Case Diagram**

The use case diagram illustrates a Construction Marketplace system where the \*Customer\* interacts with the platform to \*browse available construction materials\* provided by \*Dealers, \*\*hire skilled workers\* listed by \*Workers, and \*\*make bookings\* for construction-related services, while the \*Admin\* oversees and manages user accounts and roles within the system to ensure smooth operation and proper user access control.



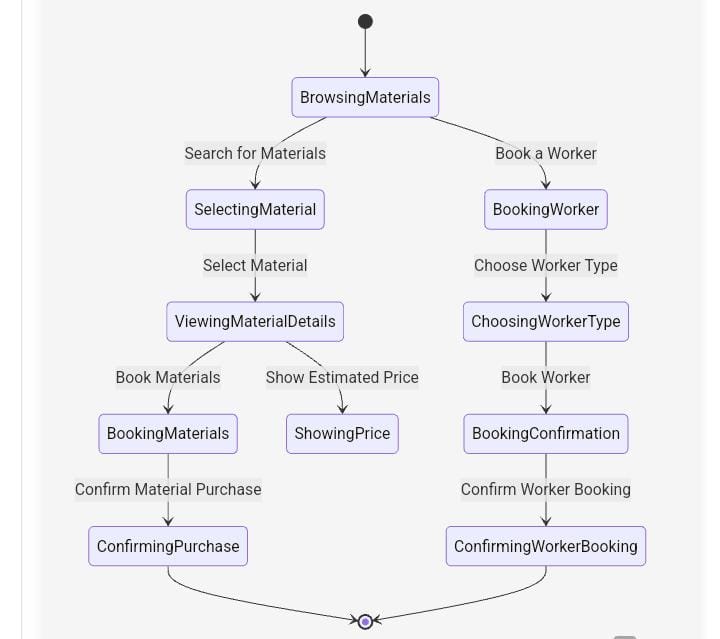
**Fig-8.3.5 Use Case Diagram**

fi

**8.3.6 State Chart Diagram**

This state chart diagram depicts a sequential process initiated by a user browsing materials. From this initial state, the user has two distinct paths they can follow. The first path involves searching for specific materials, then selecting a particular material of interest. Upon selection, the user can view the detailed information about the chosen material. Subsequently, the user has the option to either book these materials, which transitions them through a booking materials state to finally confirming their purchase. Alternatively, after viewing the material details, the user might choose to see the estimated price before making a booking decision.

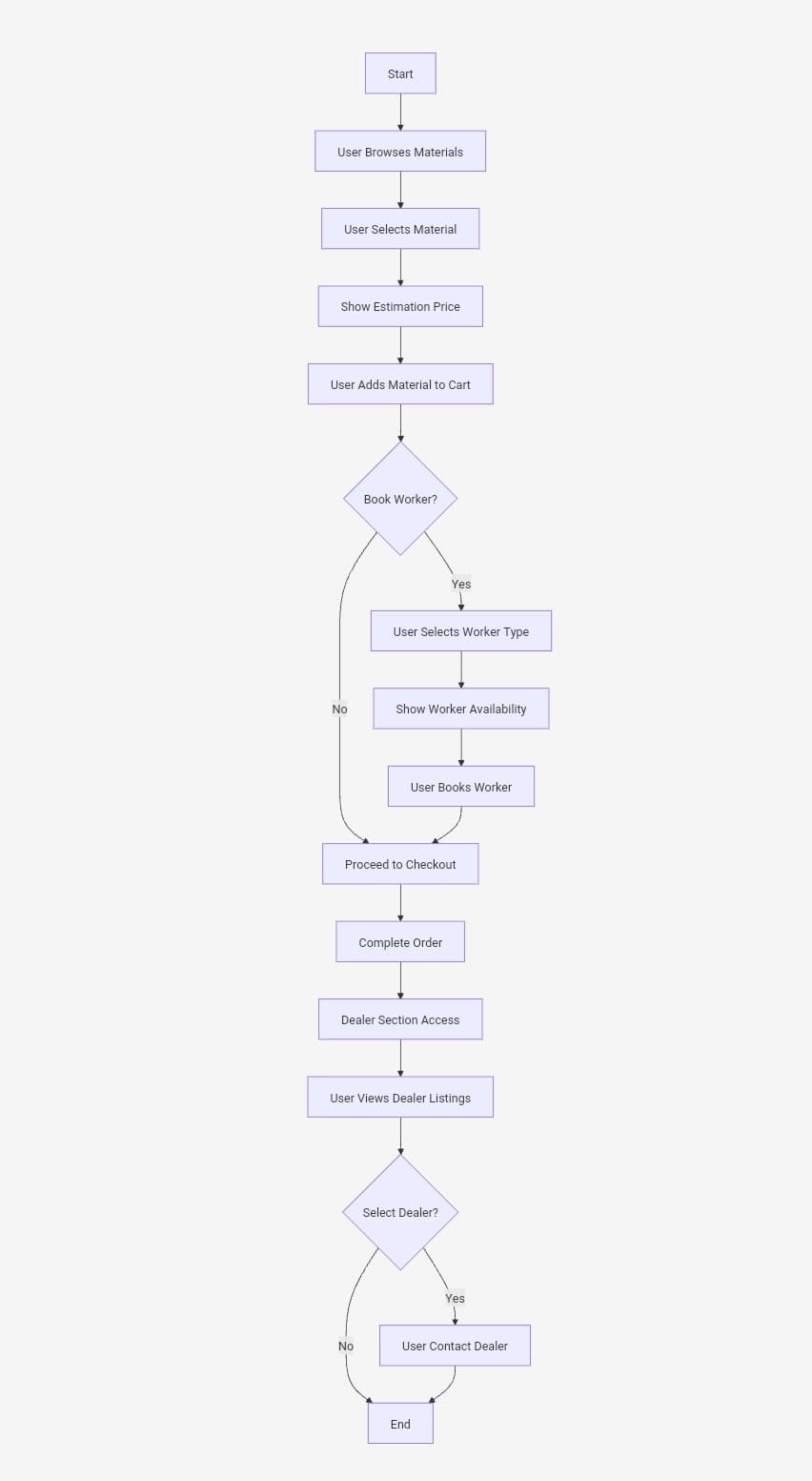
The second distinct path from the initial browsing materials state allows the user to book a worker. This involves entering a booking worker state, followed by the user choosing a specific worker type. After selecting the worker type, the user proceeds to book the worker, which leads to a booking confirmation state and ultimately confirming the worker booking. Regardless of whether the user chooses to book materials or a worker, the process converges at the end state, indicating the completion of their interaction.



**Fig-8.3.6 State Chart Diagram**

**8.3.7 Activity Diagram**

This activity diagram illustrates a workflow that involves both material selection and the potential booking of a worker, followed by a checkout process and an option to interact with dealers. It begins with a user browsing and selecting materials, then adding them to a cart. A decision point allows the user to either book a worker (involving worker type selection, availability check, and booking) or skip this step. Regardless of the worker booking decision, the user proceeds to checkout and completes their order. Subsequently, the system allows access to a dealer section where the user can view listings and choose to contact a dealer, or end the process without dealer interaction

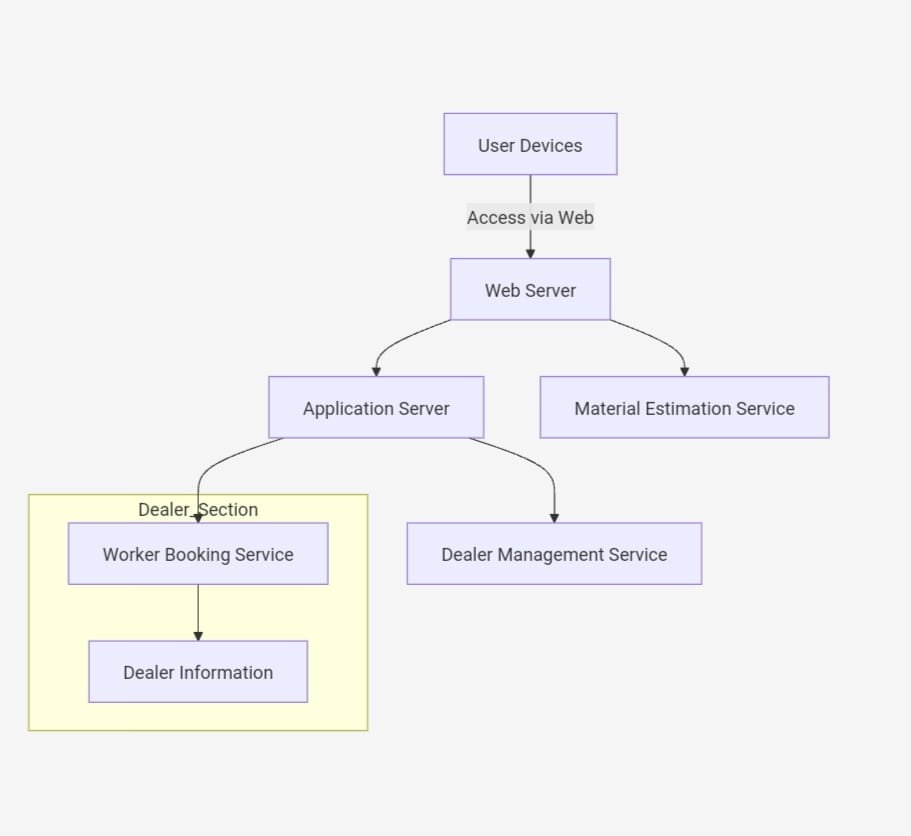


**Fig- 8.3.7 Activity Diagram**

**8.3.8 Deployment Diagram**

Deployment diagram illustrates how users and dealers interact with the system.It starts with User Devices which access the system via the Web through a Web Server.The Web Server then directs different requests to specific parts of the system. Some requests go to the Application Server, which handles the main logic of the application. Other requests are sent to the Material Estimation Service, likely for calculating material costs.

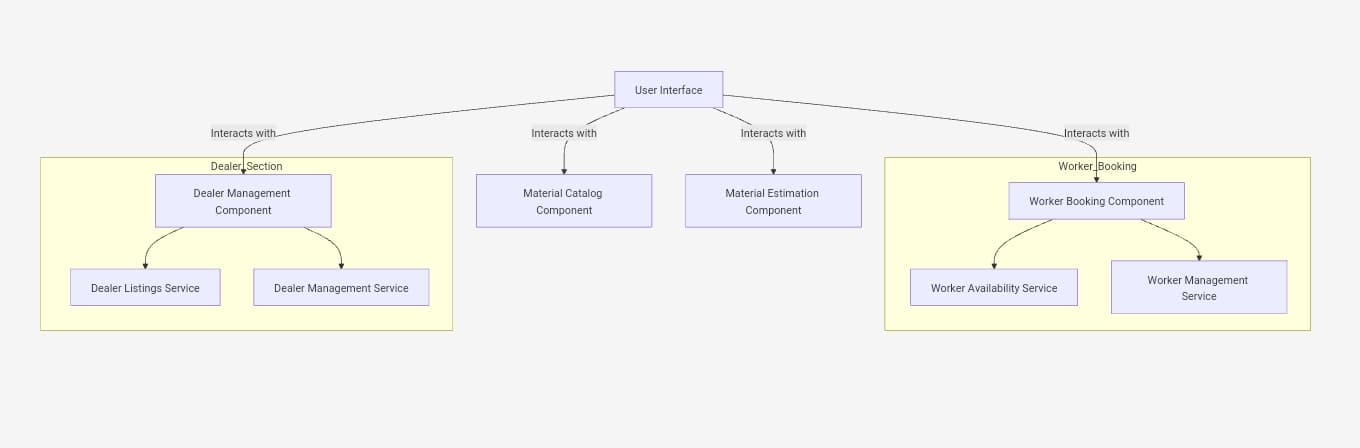
Within the Application Server, there's a dedicated Dealer Section (highlighted in yellow). This section contains the Worker Booking Service, which probably allows dealers to book workers. This service then interacts with Dealer Information, which likely stores details about the dealers.Finally, the Application Server also communicates with the Dealer Management Service, which suggests functionalities for managing dealer accounts and related operations.In simple terms, users access the system through the web, and their actions are processed by different servers. Dealers have a special section within the application to manage workers and their own information.

****

**Fig- 8.3.8 Deployment Diagram**

**8.3.9 Component Diagram**

This component diagram illustrates a system architecture where a central User Interface connects to three key functional areas. The Dealer Section allows users to manage dealer listings and related services through a dedicated management component. For Material handling, the UI interacts with components for browsing a catalog and estimating material needs. Finally, the Worker Booking section enables users to manage worker availability and related services via a booking component. This modular design, with distinct components for dealers, materials, and workers, all accessible through the user interface, promotes organization and separation of concerns within the system.



**Fig-8.3.9 Component Diagram**

## CHAPTER-9

## SOURCE CODE

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Build & Crew</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/bootstrap-icons.css">

<style>

:root {

--primary-color: #2c3e50;

--secondary-color: #3498db;

--accent-color: #e74c3c;

--background-color: #f8f9fa;

--card-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

--hover-shadow: 0 8px 15px rgba(0, 0, 0, 0.2);

--gradient-primary: linear-gradient(135deg, #3498db 0%, #2980b9 100%);

--gradient-secondary: linear-gradient(135deg, #2c3e50 0%, #34495e 100%);

}

body {

font-family: 'Segoe UI', system-ui, -apple-system, sans-serif;

background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);

min-height: 100vh;

color: var(--primary-color);

line-height: 1.6;

}

.container {

max-width: 1800px;

margin: 0 auto;

padding: 20px;

}

.hero-section {

text-align: center;

padding: 60px 0;

background: var(--gradient-secondary);

color: white;

border-radius: 0 0 30px 30px;

margin-bottom: 40px;

position: relative;

overflow: hidden;

}

.hero-section::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background:url('https://images.unsplash.com/photo-1541888946425-d81bb19240f5?ixlib=rb-1.2.1&auto=format&fit=crop&w=1950&q=80') center/cover;

opacity: 0.1;

}

h1 {

color: white;

text-align: center;

margin: 0 0 20px;

font-size: 3.5em;

font-weight: 700;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);

position: relative;

}

.hero-subtitle {

color: rgba(255, 255, 255, 0.9);

font-size: 1.2em;

max-width: 600px;

margin: 0 auto 30px;

}

.search-container {

background: rgba(255, 255, 255, 0.95);

padding: 20px 40px;

border-radius: 20px;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);

margin: -80px auto 40px;

position: relative;

z-index: 100;

backdrop-filter: blur(10px);

border: 1px solid rgba(255, 255, 255, 0.2);

max-width: 1800px;

width: 100%;

}

.search-box {

margin-bottom: 25px;

width: 100%;

overflow: hidden;

}

.search-box input {

width: calc(100% - 120px);

padding: 20px;

border: 2px solid #e0e0e0;

border-radius: 16px;

font-size: 1.2em;

transition: all 0.3s ease;

background: white;

height: 60px;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);

float: left;

}

.search-box button {

float: left;

margin-left: 10px;

}

.search-box > div:last-child {

float: right;

}

.section {

background: white;

border-radius: 20px;

padding: 30px;

margin-bottom: 40px;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.section:hover {

transform: translateY(-2px);

box-shadow: 0 12px 40px rgba(0, 0, 0, 0.15);

}

.section h2 {

color: var(--primary-color);

margin-bottom: 25px;

font-size: 2em;

font-weight: 600;

position: relative;

padding-bottom: 15px;

display: flex;

align-items: center;

gap: 10px;

}

.section h2::after {

content: '';

position: absolute;

bottom: 0;

left: 0;

width: 50px;

height: 3px;

background: var(--gradient-primary);

border-radius: 2px;

}

.material-card, .worker-card {

background: white;

border-radius: 20px;

padding: 25px;

transition: all 0.3s ease;

cursor: pointer;

box-shadow: var(--card-shadow);

height: 100%;

border: 1px solid rgba(221, 221, 221, 0.3);

display: flex;

flex-direction: column;

position: relative;

overflow: hidden;

}

.material-card::before, .worker-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gradient-primary);

}

.material-image {

width: 100%;

height: 200px;

object-fit: cover;

border-radius: 16px;

margin-bottom: 20px;

background-color: #f8f9fa;

transition: transform 0.3s ease;

}

.material-card:hover .material-image,

.worker-card:hover .worker-image {

transform: scale(1.05);

}

.material-card:hover, .worker-card:hover {

transform: translateY(-5px);

box-shadow: var(--hover-shadow);

}

.material-card h3, .worker-card h3 {

color: var(--primary-color);

margin-bottom: 15px;

font-size: 1.4em;

font-weight: 600;

}

.price {

color: var(--secondary-color);

font-weight: bold;

font-size: 1.3em;

margin: 15px 0;

display: flex;

align-items: center;

gap: 5px;

}

.rating {

color: #f1c40f;

margin: 10px 0;

display: flex;

align-items: center;

gap: 5px;

}

.badge {

padding: 8px 16px;

border-radius: 20px;

font-weight: 500;

font-size: 0.9em;

margin-bottom: 10px;

display: inline-block;

}

.badge-primary {

background: var(--gradient-primary);

color: white;

}

.badge-success {

background: linear-gradient(135deg, #2ecc71 0%, #27ae60 100%);

color: white;

}

.auth-controls {

position: fixed;

top: 20px;

right: 20px;

display: flex;

gap: 15px;

align-items: center;

z-index: 1000;

}

.auth-btn {

background: var(--gradient-primary);

color: white;

border: none;

padding: 12px 24px;

border-radius: 25px;

cursor: pointer;

text-decoration: none;

font-size: 1em;

font-weight: 500;

transition: all 0.3s ease;

box-shadow: 0 4px 8px rgba(52, 152, 219, 0.2);

display: flex;

align-items: center;

gap: 8px;

}

.auth-btn:hover {

transform: translateY(-2px);

box-shadow: 0 6px 12px rgba(52, 152, 219, 0.3);

}

.user-info {

display: flex;

align-items: center;

gap: 15px;

background: white;

padding: 10px 20px;

border-radius: 25px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.user-name {

font-weight: 500;

color: var(--primary-color);

}

.skill-tag {

background: #e9ecef;

padding: 5px 15px;

border-radius: 20px;

font-size: 0.9em;

color: var(--primary-color);

transition: all 0.3s ease;

}

.skill-tag:hover {

background: var(--secondary-color);

color: white;

}

.category-card {

background: white;

border-radius: 10px;

padding: 15px;

text-align: center;

transition: all 0.3s ease;

cursor: pointer;

box-shadow: var(--card-shadow);

border: 1px solid rgba(221, 221, 221, 0.3);

position: relative;

overflow: hidden;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.category-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gradient-primary);

}

.category-card:hover {

transform: translateY(-5px);

box-shadow: var(--hover-shadow);

}

.category-icon {

font-size: 1.5em;

color: var(--secondary-color);

margin-bottom: 8px;

width: 45px;

height: 45px;

display: flex;

align-items: center;

justify-content: center;

background: rgba(52, 152, 219, 0.1);

border-radius: 10px;

}

.category-card h3 {

color: var(--primary-color);

font-size: 1em;

font-weight: 600;

margin-bottom: 3px;

}

.category-card p {

color: #666;

font-size: 0.75em;

margin: 0;

}

.categories-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));

gap: 15px;

width: 100%;

}

.worker-card {

background: white;

border-radius: 20px;

padding: 25px;

transition: all 0.3s ease;

cursor: pointer;

box-shadow: var(--card-shadow);

height: 100%;

border: 1px solid rgba(221, 221, 221, 0.3);

display: flex;

flex-direction: column;

position: relative;

overflow: hidden;

}

.worker-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gradient-primary);

}

.worker-card:hover {

transform: translateY(-5px);

box-shadow: var(--hover-shadow);

}

.worker-image {

width: 100%;

height: 200px;

object-fit: cover;

border-radius: 16px;

margin-bottom: 20px;

background-color: #f8f9fa;

transition: transform 0.3s ease;

}

.worker-card:hover .worker-image {

transform: scale(1.05);

}

.worker-card h3 {

color: var(--primary-color);

margin-bottom: 15px;

font-size: 1.4em;

font-weight: 600;

}

.availability-badge {

position: absolute;

top: 20px;

right: 20px;

padding: 8px 16px;

border-radius: 20px;

font-weight: 500;

font-size: 0.9em;

z-index: 1;

}

.availability-badge.available {

background: linear-gradient(135deg, #2ecc71 0%, #27ae60 100%);

color: white;

}

.availability-badge.unavailable {

background: linear-gradient(135deg, #e74c3c 0%, #c0392b 100%);

color: white;

}

.worker-rating {

display: flex;

align-items: center;

gap: 5px;

color: #f1c40f;

margin: 10px 0;

}

.worker-info {

display: flex;

flex-direction: column;

gap: 10px;

margin: 15px 0;

}

.worker-info p {

margin: 0;

display: flex;

align-items: center;

gap: 8px;

}

.worker-info i {

color: var(--secondary-color);

font-size: 1.1em;

}

.worker-skills {

display: flex;

flex-wrap: wrap;

gap: 8px;

margin-top: 15px;

}

.worker-skill-tag {

background: rgba(52, 152, 219, 0.1);

color: var(--secondary-color);

padding: 6px 12px;

border-radius: 15px;

font-size: 0.85em;

transition: all 0.3s ease;

}

.worker-skill-tag:hover {

background: var(--secondary-color);

color: white;

}

.worker-category-card {

background: white;

border-radius: 8px;

padding: 10px;

text-align: center;

transition: all 0.3s ease;

cursor: pointer;

box-shadow: var(--card-shadow);

border: 1px solid rgba(221, 221, 221, 0.3);

position: relative;

overflow: hidden;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.worker-category-icon {

font-size: 1.2em;

color: var(--secondary-color);

margin-bottom: 5px;

width: 35px;

height: 35px;

display: flex;

align-items: center;

justify-content: center;

background: rgba(52, 152, 219, 0.1);

border-radius: 8px;

}

.worker-category-card h3 {

color: var(--primary-color);

font-size: 0.9em;

font-weight: 600;

margin-bottom: 2px;

}

.worker-category-card p {

color: #666;

font-size: 0.7em;

margin: 0;

}

.worker-categories-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(120px, 1fr));

gap: 10px;

width: 100%;

margin-bottom: 30px;

}

.review-modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.5);

z-index: 1000;

align-items: center;

justify-content: center;

}

.review-content {

background: white;

padding: 30px;

border-radius: 20px;

width: 90%;

max-width: 500px;

position: relative;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);

}

.close-review {

position: absolute;

top: 15px;

right: 15px;

font-size: 1.5em;

cursor: pointer;

color: var(--primary-color);

}

.rating-input {

display: flex;

gap: 10px;

margin: 20px 0;

}

.rating-input i {

font-size: 1.5em;

color: #ddd;

cursor: pointer;

transition: all 0.3s ease;

}

.rating-input i.active {

color: #f1c40f;

}

.review-textarea {

width: 100%;

padding: 15px;

border: 2px solid #e0e0e0;

border-radius: 10px;

margin: 15px 0;

resize: vertical;

min-height: 100px;

font-family: inherit;

}

.review-textarea:focus {

outline: none;

border-color: var(--secondary-color);

}

.submit-review {

background: var(--gradient-primary);

color: white;

border: none;

padding: 12px 24px;

border-radius: 25px;

cursor: pointer;

font-size: 1em;

font-weight: 500;

transition: all 0.3s ease;

width: 100%;

}

.submit-review:hover {

transform: translateY(-2px);

box-shadow: 0 6px 12px rgba(52, 152, 219, 0.3);

}

.reviews-section {

margin-top: 20px;

padding-top: 20px;

border-top: 1px solid #eee;

}

.review-item {

background: #f8f9fa;

padding: 15px;

border-radius: 10px;

margin-bottom: 15px;

}

.review-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 10px;

}

.reviewer-name {

font-weight: 600;

color: var(--primary-color);

}

.review-date {

color: #666;

font-size: 0.9em;

}

.review-rating {

color: #f1c40f;

margin: 5px 0;

}

.review-text {

color: #333;

line-height: 1.5;

}

@media (max-width: 768px) {

.container {

padding: 15px;

}

h1 {

font-size: 2.5em;

}

.search-container {

padding: 20px;

margin: -20px auto 30px;

}

.section {

padding: 20px;

}

.material-card, .worker-card {

padding: 20px;

}

}

.navbar {

background-color: #f8f9fa;

padding: 1rem;

margin-bottom: 1rem;

}

.navbar-brand {

color: var(--primary-color);

text-decoration: none;

font-size: 1.25rem;

font-weight: 600;

}

</style>

</head>

<body>

<div class="hero-section">

<div class="container">

<h1>Build & Crew</h1>

<p class="hero-subtitle">Your one-stop platform for construction materials and skilled workers</p>

</div>

</div>

<div class="container">

<div class="search-container">

<div class="search-box d-flex justify-content-between align-items-center">

<div class="d-flex flex-grow-1 me-3">

<input type="text" id="searchInput" placeholder="Search for materials or workers...">

<button class="auth-btn ms-2" onclick="performSearch()">

<i class="bi bi-search"></i> Search

</button>

</div>

<div class="d-flex align-items-center">

<a href="cart.html" class="nav-link me-3">

<i class="bi bi-cart3"></i>

<span id="cartCount" class="badge bg-primary">0</span>

</a>

<div id="userSection">

<div id="userInfo" style="display: none;" class="user-info">

<span class="user-name"></span>

<button onclick="logout()" class="auth-btn">

<i class="bi bi-box-arrow-right"></i> Logout

</button>

</div>

<div id="authButtons">

<a href="login.html" class="auth-btn">

<i class="bi bi-box-arrow-in-right"></i> Login

</a>

</div>

</div>

</div>

</div>

</div>

<div class="section">

<h2><i class="bi bi-grid"></i> Categories</h2>

<div class="row" id="categoriesContainer">

<!-- Categories will be loaded here -->

</div>

</div>

<div class="section">

<h2><i class="bi bi-box"></i> Popular Materials</h2>

<div class="row" id="materialsContainer">

<!-- Materials will be loaded here -->

</div>

</div>

<div class="section">

<h2><i class="bi bi-people"></i> Workers</h2>

<div class="worker-categories-grid" id="workerCategoriesContainer">

<!-- Worker categories will be loaded here -->

</div>

<div class="row" id="workersContainer">

<!-- Workers will be loaded here -->

</div>

</div>

</div>

<!-- Review Modal -->

<div class="review-modal" id="reviewModal">

<div class="review-content">

<span class="close-review" onclick="closeReviewModal()">&times;</span>

<h3>Write a Review</h3>

<div class="rating-input" id="ratingInput">

<i class="bi bi-star" onclick="setRating(1)"></i>

<i class="bi bi-star" onclick="setRating(2)"></i>

<i class="bi bi-star" onclick="setRating(3)"></i>

<i class="bi bi-star" onclick="setRating(4)"></i>

<i class="bi bi-star" onclick="setRating(5)"></i>

</div>

<textarea class="review-textarea" id="reviewText" placeholder="Write your review here..."></textarea>

<button class="submit-review" onclick="submitReview()">Submit Review</button>

</div>

</div>

<script>

let allMaterials = [];

let allWorkers = [];

let currentRating = 0;

let currentItemId = null;

let currentItemType = null; // 'material' or 'worker'

// Fetch and display materials

async function loadMaterials() {

try {

const response = await fetch('/api/materials');

const materials = await response.json();

allMaterials = materials; // Store all materials

displayMaterials(materials);

displayPopularMaterials(materials);

} catch (error) {

console.error('Error loading materials:', error);

displaySampleMaterials();

}

}

function displayMaterials(materials) {

const container = document.getElementById('categoriesContainer');

// Get unique categories

const categories = [...new Set(materials.map(material => material.category))];

// Create HTML for each category

container.innerHTML = `

<div class="categories-grid">

${categories.map(category => `

<div class="category-card" onclick="window.location.href='category-materials.html?category=${encodeURIComponent(category)}'">

<div class="category-icon">

<i class="bi bi-box-seam"></i>

</div>

<h3>${category}</h3>

<p>View Materials</p>

</div>

`).join('')}

</div>

`;

// Display popular materials

const materialsContainer = document.getElementById('materialsContainer');

const popularMaterials = materials.length > 8

? materials.sort(() => 0.5 - Math.random()).slice(0, 8)

: materials;

materialsContainer.innerHTML = popularMaterials.map(material => `

<div class="col-md-4 col-lg-3 mb-4">

<div class="material-card" onclick="window.location.href='material.html?id=${material.id}'">

<img src="${material.image || 'https://via.placeholder.com/400x300/f8f9fa/2c3e50.png?text=Material+Image'}"

alt="${material.name}"

class="material-image">

<h3>${material.name}</h3>

<p class="text-muted">${material.description}</p>

<div class="price">₹${material.price} per ${material.unit}</div>

<div class="rating">

<i class="bi bi-star-fill"></i> ${material.rating || 'Not rated yet'}

</div>

<button class="auth-btn" style="margin-top: 15px;" onclick="event.stopPropagation(); window.location.href='reviews.html?id=${material.id}&type=material'">

<i class="bi bi-star"></i> Rate & Review

</button>

${material.reviews && material.reviews.length > 0 ? `

<div class="reviews-section">

<h4>Recent Reviews</h4>

${material.reviews.slice(0, 2).map(review => `

<div class="review-item">

<div class="review-header">

<span class="reviewer-name">${review.userName}</span>

<span class="review-date">${new Date(review.date).toLocaleDateString()}</span>

</div>

<div class="review-rating">

${'<i class="bi bi-star-fill"></i>'.repeat(review.rating)}

</div>

<p class="review-text">${review.text}</p>

</div>

`).join('')}

</div>

` : ''}

</div>

</div>

`).join('');

}

function displayPopularMaterials(materials) {

const container = document.getElementById('materialsContainer');

if (!materials || materials.length === 0) {

displaySampleMaterials();

return;

}

// Get 8 random materials or all if less than 8

const popularMaterials = materials.length > 8

? materials.sort(() => 0.5 - Math.random()).slice(0, 8)

: materials;

container.innerHTML = popularMaterials.map(material => `

<div class="col-md-4 col-lg-3 mb-4">

<div class="material-card" onclick="window.location.href='material.html?id=${material.id}'">

<img src="${material.image || 'https://via.placeholder.com/400x300/f8f9fa/2c3e50.png?text=Material+Image'}"

alt="${material.name}"

class="material-image">

<h3>${material.name}</h3>

<p class="text-muted">${material.description}</p>

<div class="price">₹${material.price} per ${material.unit}</div>

</div>

</div>

`).join('');

}

function displaySampleMaterials() {

const container = document.getElementById('materialsContainer');

const sampleMaterials = [

{

id: 1,

name: "Premium Cement",

description: "High-quality cement for strong foundations",

price: 450,

unit: "bag",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",

rating: 4.5,

reviews: [

{

userName: "Rahul Sharma",

date: "2024-03-15",

rating: 5,

text: "Excellent quality cement. Very strong and durable."

},

{

userName: "Priya Patel",

date: "2024-03-10",

rating: 4,

text: "Good quality but a bit expensive."

}

]

},

{

id: 2,

name: "Steel Bars",

description: "TMT steel bars for reinforced concrete",

price: 65000,

unit: "ton",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",

rating: 4.8,

reviews: [

{

userName: "Amit Kumar",

date: "2024-03-12",

rating: 5,

text: "Best quality TMT bars I've used."

}

]

},

{

id: 3,

name: "Bricks",

description: "High-quality red bricks for construction",

price: 8,

unit: "piece",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

},

{

id: 4,

name: "Sand",

description: "Fine quality construction sand",

price: 1200,

unit: "truck",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

},

{

id: 5,

name: "Tiles",

description: "Ceramic floor tiles",

price: 45,

unit: "sq.ft",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

},

{

id: 6,

name: "Plywood",

description: "High-grade plywood sheets",

price: 1200,

unit: "sheet",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

},

{

id: 7,

name: "Paint",

description: "Premium exterior paint",

price: 1200,

unit: "liter",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

},

{

id: 8,

name: "PVC Pipes",

description: "Durable PVC plumbing pipes",

price: 150,

unit: "meter",

image: "https://images.unsplash.com/photo-1581094794329-c8112c4e3f1f?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60"

}

];

container.innerHTML = sampleMaterials.map(material => `

<div class="col-md-4 col-lg-3 mb-4">

<div class="material-card" onclick="window.location.href='material.html?id=${material.id}'">

<img src="${material.image}"

alt="${material.name}"

class="material-image">

<h3>${material.name}</h3>

<p class="text-muted">${material.description}</p>

<div class="price">₹${material.price} per ${material.unit}</div>

<div class="rating">

<i class="bi bi-star-fill"></i> ${material.rating || 'Not rated yet'}

</div>

<button class="auth-btn" style="margin-top: 15px;" onclick="event.stopPropagation(); window.location.href='reviews.html?id=${material.id}&type=material'">

<i class="bi bi-star"></i> Rate & Review

</button>

${material.reviews && material.reviews.length > 0 ? `

<div class="reviews-section">

<h4>Recent Reviews</h4>

${material.reviews.slice(0, 2).map(review => `

<div class="review-item">

<div class="review-header">

<span class="reviewer-name">${review.userName}</span>

<span class="review-date">${new Date(review.date).toLocaleDateString()}</span>

</div>

<div class="review-rating">

${'<i class="bi bi-star-fill"></i>'.repeat(review.rating)}

</div>

<p class="review-text">${review.text}</p>

</div>

`).join('')}

</div>

` : ''}

</div>

</div>

`).join('');

}

// Fetch and display workers

async function loadWorkers() {

try {

const response = await fetch('/api/workers');

const workers = await response.json();

allWorkers = workers; // Store all workers

console.log('Loaded workers:', workers); // Debug log

displayWorkerCategories(workers);

displayWorkers(workers);

} catch (error) {

console.error('Error loading workers:', error);

displaySampleWorkers();

}

}

function displayWorkerCategories(workers) {

const container = document.getElementById('workerCategoriesContainer');

// Get unique specialties

const specialties = [...new Set(workers.map(worker => worker.specialty))];

// Create HTML for each category

container.innerHTML = `

${specialties.map(specialty => `

<div class="worker-category-card" onclick="filterWorkers('${specialty}')">

<div class="worker-category-icon">

<i class="bi bi-person-badge"></i>

</div>

<h3>${specialty}</h3>

<p>View Workers</p>

</div>

`).join('')}

`;

}

function displayWorkers(workers) {

const container = document.getElementById('workersContainer');

if (!container) {

console.error('Workers container not found');

return;

}

if (!workers || workers.length === 0) {

container.innerHTML = '<div class="col-12 text-center"><p>No workers available at the moment.</p></div>';

return;

}

container.innerHTML = workers.map(worker => `

<div class="col-md-4 col-lg-3 mb-4">

<div class="worker-card" onclick="window.location.href='worker.html?id=${worker.id}'">

<div class="availability-badge ${worker.available ? 'available' : 'unavailable'}">

${worker.available ? 'Available' : 'Unavailable'}

</div>

<h3>${worker.name || 'Unnamed Worker'}</h3>

<div class="badge badge-primary">${worker.specialty || 'General Worker'}</div>

<div class="worker-rating">

<i class="bi bi-star-fill"></i> ${worker.rating || 'Not rated yet'}

</div>

<div class="worker-info">

<p><i class="bi bi-briefcase"></i> <strong>Experience:</strong> ${worker.experience || 'Not specified'}</p>

<p><i class="bi bi-cash"></i> <strong>Rate:</strong> ₹${worker.rate || 'Not specified'} per day</p>

<p><i class="bi bi-geo-alt"></i> <strong>Location:</strong> ${worker.location || 'Not specified'}</p>

</div>

<div class="worker-skills">

${(worker.skills && worker.skills.length > 0)

? worker.skills.map(skill => <span class="worker-skill-tag">${skill}</span>).join('')

: '<span class="worker-skill-tag">No skills listed</span>'}

</div>

${worker.reviews && worker.reviews.length > 0 ? `

<div class="reviews-section">

<h4>Recent Reviews</h4>

${worker.reviews.slice(0, 2).map(review => `

<div class="review-item">

<div class="review-header">

<span class="reviewer-name">${review.userName}</span>

<span class="review-date">${new Date(review.date).toLocaleDateString()}</span>

</div>

<div class="review-rating">

${'<i class="bi bi-star-fill"></i>'.repeat(review.rating)}

</div>

<p class="review-text">${review.text}</p>

</div>

`).join('')}

</div>

` : ''}

</div>

</div>

`).join('');

}

function filterWorkers(specialty) {

const filteredWorkers = allWorkers.filter(worker => worker.specialty === specialty);

displayWorkers(filteredWorkers);

}

function displaySampleWorkers() {

const sampleWorkers = [

{

id: 1,

name: "Rajesh Kumar",

specialty: "Mason",

available: true,

rating: 4.8,

experience: "15 years",

rate: 1200,

location: "Mumbai",

skills: ["Brickwork", "Plastering", "Tiling"],

reviews: [

{

userName: "Vikram Singh",

date: "2024-03-14",

rating: 5,

text: "Excellent workmanship and very professional."

},

{

userName: "Anita Desai",

date: "2024-03-08",

rating: 4,

text: "Good worker, completed the job on time."

}]

},

{

id: 2,

name: "Amit Singh",

specialty: "Carpenter",

available: true,

rating: 4.6,

experience: "12 years",

rate: 1000,

location: "Delhi",

skills: ["Furniture", "Doors", "Windows"],

reviews: [

{

userName: "Rahul Verma",

date: "2024-03-10",

rating: 5,

text: "Skilled carpenter, very satisfied with the work."

}

]

}

];

displayWorkerCategories(sampleWorkers);

displayWorkers(sampleWorkers);

}

// Search functionality

function performSearch() {

const searchTerm = document.getElementById('searchInput').value.toLowerCase();

// Filter materials

const filteredMaterials = allMaterials.filter(material =>

material.name.toLowerCase().includes(searchTerm) ||

material.category.toLowerCase().includes(searchTerm) ||

material.description.toLowerCase().includes(searchTerm) ||

material.supplier?.toLowerCase().includes(searchTerm)

);

displayMaterials(filteredMaterials)

const filteredWorkers = allWorkers.filter(worker =>

(worker.name && worker.name.toLowerCase().includes(searchTerm)) ||

(worker.specialty && worker.specialty.toLowerCase().includes(searchTerm)) ||

(worker.experience && worker.experience.toLowerCase().includes(searchTerm)) ||

(worker.location && worker.location.toLowerCase().includes(searchTerm)) ||

(worker.skills && worker.skills.some(skill => skill.toLowerCase().includes(searchTerm)))

);

displayWorkers(filteredWorkers);

}

// Add event listener for real-time search

document.getElementById('searchInput').addEventListener('input', performSearch);

// Check authentication status and update UI

function checkAuth() {

const token = localStorage.getItem('token');

const userInfo = document.getElementById('userInfo');

const authButtons = document.getElementById('authButtons');

if (token) {

try {

const userString = localStorage.getItem('user');

const user = userString ? JSON.parse(userString) : {};

if (user && user.username) {

userInfo.style.display = 'flex';

authButtons.style.display = 'none';

userInfo.querySelector('.user-name').textContent = Welcome, ${user.username};

} else {

// Invalid user data, clear storage and show login

localStorage.removeItem('token');

localStorage.removeItem('user');

userInfo.style.display = 'none';

authButtons.style.display = 'flex';

}

} catch (error) {

console.error('Error parsing user data:', error);

// Clear invalid data

localStorage.removeItem('token');

localStorage.removeItem('user');

userInfo.style.display = 'none';

authButtons.style.display = 'flex';

}

} else {

userInfo.style.display = 'none';

authButtons.style.display = 'flex';

}

}

// Logout function

function logout() {

localStorage.removeItem('token');

localStorage.removeItem('user');

window.location.reload();

}

document.addEventListener('DOMContentLoaded', () => {

checkAuth();

loadMaterials();

loadWorkers();

});

function openReviewModal(itemId, itemType) {

currentItemId = itemId;

currentItemType = itemType;

document.getElementById('reviewModal').style.display = 'flex';

}

function closeReviewModal() {

document.getElementById('reviewModal').style.display = 'none';

currentRating = 0;

document.getElementById('reviewText').value = '';

updateRatingStars();

}

function setRating(rating) {

currentRating = rating;

updateRatingStars();

}

function updateRatingStars() {

const stars = document.querySelectorAll('#ratingInput i');

stars.forEach((star, index) => {

star.classList.toggle('active', index < currentRating);

});

}

async function submitReview() {

if (currentRating === 0) {

alert('Please select a rating');

return;

}

const reviewText = document.getElementById('reviewText').value.trim();

if (!reviewText) {

alert('Please write a review');

return;

}

try {

const token = localStorage.getItem('token');

if (!token) {

alert('Please login to submit a review');

return;

}

const response = await fetch('/api/reviews', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Authorization': Bearer ${token}

},

body: JSON.stringify({

itemId: currentItemId,

itemType: currentItemType,

rating: currentRating,

text: reviewText

})

});

if (response.ok) {

alert('Review submitted successfully');

closeReviewModal();

if (currentItemType === 'material') {

loadMaterials();

} else {

loadWorkers();

}

} else {

throw new Error('Failed to submit review');

}

} catch (error) {

console.error('Error submitting review:', error);

alert('Failed to submit review. Please try again.');

}

}

// Add back the updateCartCount function and its event listener

function updateCartCount() {

const cart = JSON.parse(localStorage.getItem('cart') || '[]');

document.getElementById('cartCount').textContent = cart.length;

}

document.addEventListener('DOMContentLoaded', () => {

checkAuth();

loadMaterials();

loadWorkers();

updateCartCount();

});

</script>

</body>

</html>

**CHAPTER-10**

**IMPLEMENTATION**

The implementation phase transforms the system design into a working solution through code development, UI structuring, data integration, and testing. This chapter describes the steps taken to implement the core functionalities of the construction marketplace website including material procurement, worker booking, cost estimation, and data handling using JSON.

**10.1 Technology Stack**

* **Frontend:** HTML5, CSS3, JavaScript
* **Backend / Logic Handling:** Python,API
* **Data Storage:** JSON files (instead of a database)
* **Hosting:** Localhost or lightweight deployment environment (e.g., XAMPP, Python HTTP server)

**10.2 Material Procurement Implementation**

* A dedicated HTML page lists various construction materials (cement, sand, bricks, electrical tools).
* JavaScript functions allow users to filter, search, and compare materials based on price and availability.
* Data about materials (name, price, stock, unit, category) is stored in structured JSON files and fetched dynamically.
* On selection, materials can be added to a cart with quantity options, and the total price updates accordingly.

**10.3 Skilled Worker Booking Implementation**

* Workers are categorized by skill (e.g., plumber, carpenter, electrician) and displayed with their profile cards.
* Each card includes the worker’s name, hourly rate, experience, and location.
* Users can filter workers by location and skill, and book them for a selected number of hours/days.
* Booking details (user ID, worker ID, schedule) are recorded in a JSON file to simulate persistence.

**10.4 User Interface Development**

* A clean and responsive UI is developed using HTML and CSS.
* Navigation bar includes links to: Home, Materials, Workers, Estimation, Bookings.
* UI elements such as buttons, forms, and cards are styled for usability and accessibility.
* JavaScript ensures interactivity (form validation, price updates, selection logic).

**10.5 JSON-Based Data Handling**

* All data (materials, workers, bookings, users) are stored in separate JSON files.
* Python scripts are used to read, write, and update JSON data with error handling.
* Example JSON structures:
* materials.json: holds item name, price, stock, unit
* workers.json: holds name, skill, location, hourly rate
* bookings.json: stores user-booking combinations
* This file-based system mimics database operations for smaller scale applications.

**10.6 User Management Implementation**

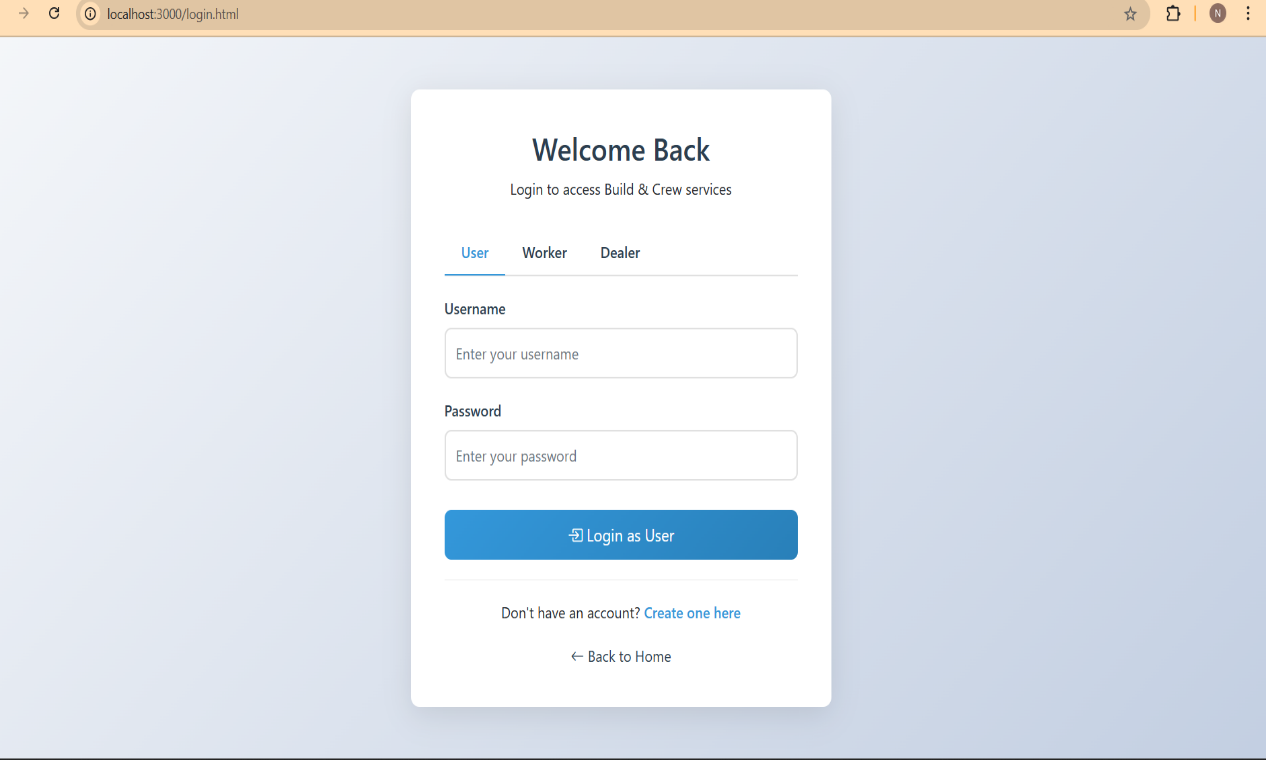
* User registration and login systems are built using HTML forms and backend Python scripts.
* Credentials and user data are saved in a users.json file.
* Logged-in users can manage their bookings, check estimates, and view their history.

**10.7 Integration and Testing**

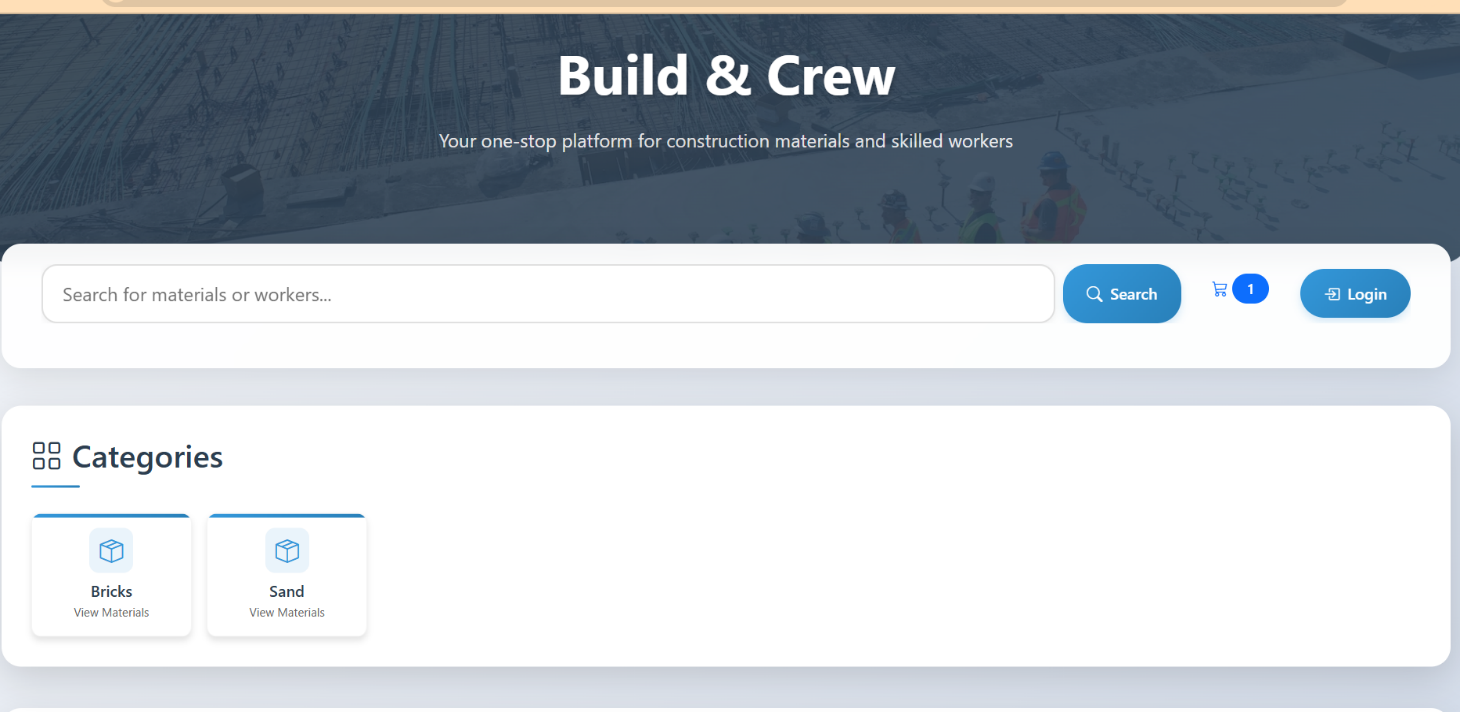
* After developing individual modules, integration was done to connect them end-to-end.
* Example flow: User selects materials → adds to cart → books a worker → sees cost estimate.
* Manual testing and validation were performed to ensure each function works as expected.

**CHAPTER-11**

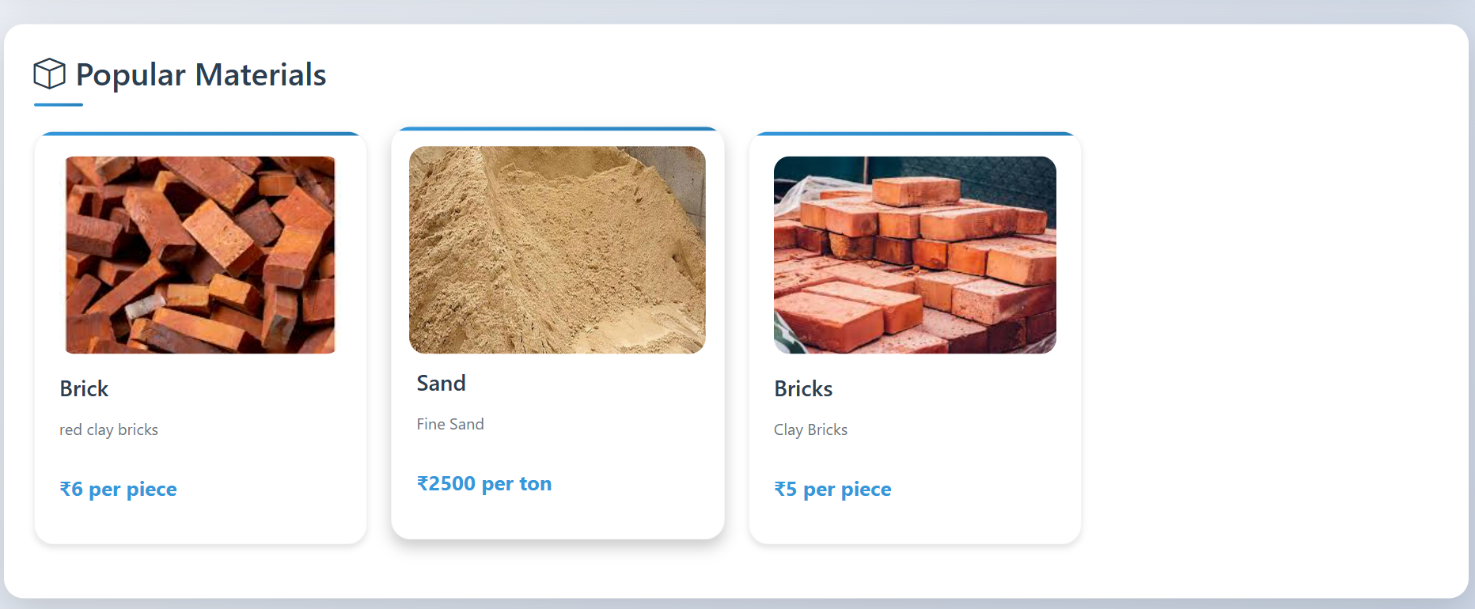
**RESULTS**

****

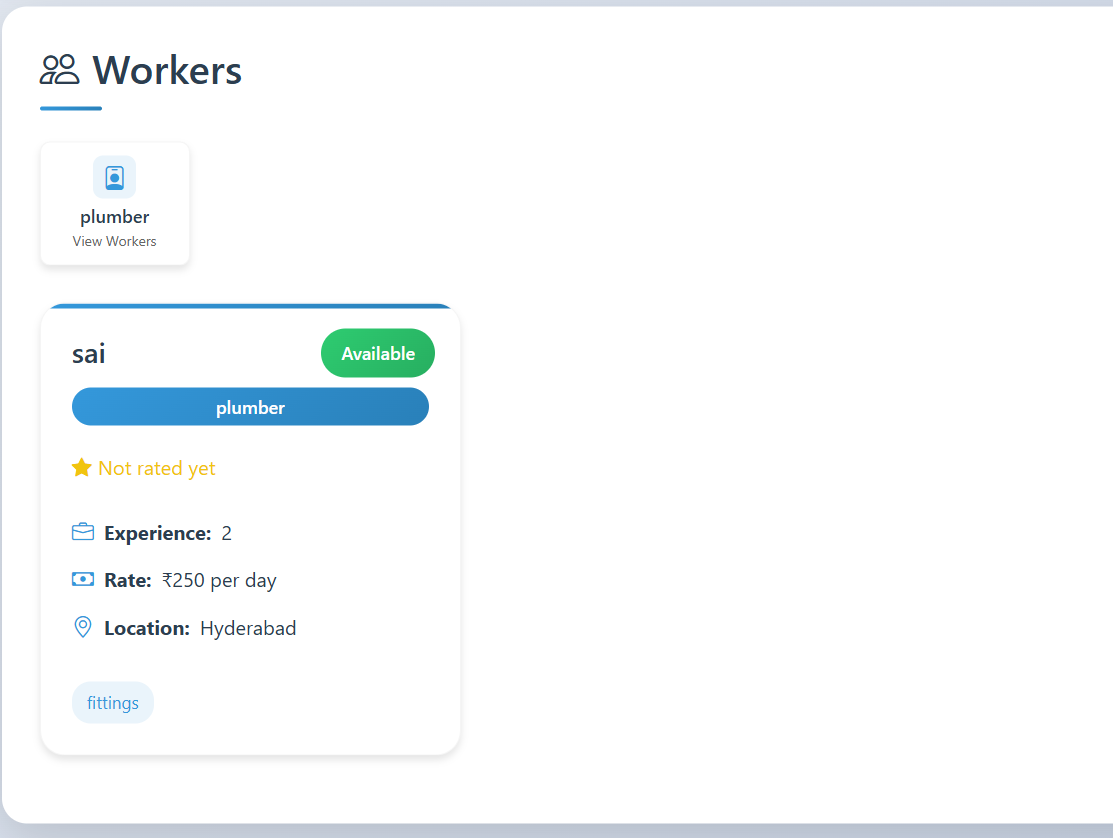
**Fig-10.1 Login Page**



**Fig:10.2 Home Page**

****

**Fig-10.3 Main page**

****

**Fig-10.3 Main page**

**CHAPTER-12**

**TESTING**

The testing phase is crucial to ensure the functionality, performance, and reliability of the construction marketplace website. Multiple testing strategies were applied to different parts of the system to validate its behavior, especially since JSON files were used for data storage instead of a database.

**12.1 Unit Testing**

Each module of the system—including the material catalog, worker booking section, and cost estimation tool—was tested individually. Special attention was given to data loading from JSON files and correct functionality under edge cases like zero quantity or invalid entries.

**12.2 Integration Testing**

Once the modules were confirmed to work in isolation, integration testing was conducted to ensure smooth data flow between modules. For example, material selection directly influenced cost estimation, and booking updates were correctly written to JSON files.

**12.3 System Testing**

The entire website was tested in a local deployment environment to simulate real user behavior. The complete user journey—from selecting materials to booking workers and viewing the total estimate—was verified for logical flow and system stability.

**12.4 UI Testing**

User interface testing ensured that the layout was responsive and usable on various devices. Forms, buttons, filters, and navigation links were checked for proper functioning.

**12.5 Validation Testing**

Input fields were tested for constraints like required values, valid numeric entries, and formatting. Invalid data such as letters in quantity fields or missing inputs were handled with error messages.

**12.6 Error & Exception Handling**

Special test cases simulated scenarios like missing JSON files or corrupted data. The system displayed fallback messages and continued running without crashing

## CHAPTER-13

**CONCLUSION AND FUTURE SCOPE**

**13.1 Conclusion**

The Construction Marketplace Website successfully bridges the gap between material procurement and skilled labor booking in the construction industry. By providing an integrated platform where users can purchase a wide range of construction materials, book certified workers, and receive real-time cost estimations, the system streamlines the construction project planning process. It eliminates the inefficiencies of traditional methods, improves transparency with clear pricing, and offers a user-friendly interface to enhance the overall user experience. Furthermore, it supports suppliers and workers by expanding their reach to a broader customer base, helping them grow their businesses. This platform ultimately promotes convenience, reliability, and better project management for individuals, contractors, and companies, revolutionizing the way construction needs are addressed in a digital era.

**13.2 Future Scope**

The Construction Marketplace Website has great potential for future growth and expansion. In the coming years, the platform can introduce features like AI-based project planning tools that suggest materials and workers automatically based on user needs. It can also expand to include real-time tracking of material deliveries and worker schedules to improve transparency.

Adding features like loyalty rewards, discounts, and subscription plans for regular users can further boost customer engagement. Integration of Augmented Reality (AR) for material visualization and virtual project walkthroughs can enhance the user experience.

In addition, partnerships with local and international suppliers and certification programs for workers can improve the quality and trust in services. Expanding the platform to mobile apps and supporting multiple languages will help reach a wider audience, including users in rural and remote areas. Overall, the platform has the scope to evolve into a complete ecosystem for managing small to large construction projects seamlessly and efficiently.

**CHAPTER-14**

**BIBILIOGRAPHY**

1. Authors -Alaa Alshwaikh, Jeehan AlShabani, Kawthar Tooq, Ehab Juma Adwan, Yusuf Almalki “A Mobile application prototype of materials procurement for the Bahraini construction industry”, 08 April 2022.
2. Davis, P., Love, P., & Baccarini, D. (2017). *Building Procurement Methods: An Overview.* Construction Management and Economics.
3. Zhao, X., Hwang, B.-G., & Low, S. P. (2018). *Integrating Labor Services and Material Procurement for Construction Efficiency.* Journal of Construction Engineering and Management.
4. Brown, K., & Yates, D. (2015). *Cost Estimation and Budgeting in Construction Projects.* International Journal of Project Management.
5. Li, H., Arditi, D., & Wang, Z. (2019). *Trust in E-commerce Construction Marketplaces.* Journal of Civil Engineering and Management.
6. Chen, X., Lu, W., & Xue, F. (2020). *Challenges in Developing Integrated Construction Platforms.* Automation in Construction.
7. Miller, T., & Jackson, P. (2018). *Mobile Solutions for Construction Management.* Journal of Information Technology in Construction.
8. Amazon Official Website. <https://www.amazon.com>
9. Urban Company (formerly UrbanClap) Official Website. <https://www.urbancompany.com>
10. TaskRabbit Official Website. <https://www.taskrabbit.com>
11. Wang, J., Wu, P., & Wang, X. (2020). *Real-Time Cost Estimation Tools in Construction.* Building Research & Information.