**Project Title:-Construction Cost Estimation**

* **Team Members:**

1. Harikishore Radhakrishnaraja
2. Jayesh Mahajan

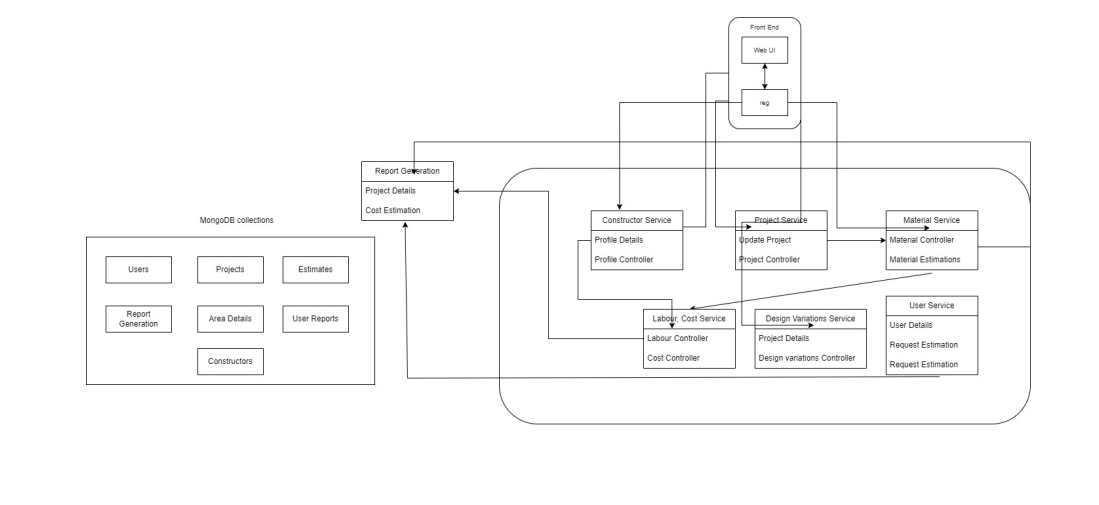
* **Problem Statement:-** : Homeowners and contractors often struggle with accurately estimating the costs for home renovations or new constructions. They rely on manual calculations that often result in errors, budget overruns, and project delays. They need a solution that allows them to input project specifications and get detailed cost estimates, including materials, labor, and design variations, to better plan their finances and timelines

1. **Understanding of Problem Statement**
2. **Problem Context:**

This application serves as a bridge between users seeking construction services and constructors providing cost estimates. Users can create requests detailing their construction needs, while constructors can view these requests and generate estimates based on various parameters, including material quality

1. **Key Requirements Identified:**
2. Request Creation: Users can create requests for cost estimations, including project details like area, type of construction, and preferred materials.
3. Constructor Dashboard: Constructors can view a list of user requests in their dashboard. o Constructors can filter requests based on location, project type, and other criteria.
4. Cost Estimation: Constructors can enter cost estimates based on user requests. Constructors can choose material quality (basic, premium, moderate) to calculate costs.
5. Report Generation: After completing the estimation, constructors can generate and submit reports to users. o Reports should include detailed cost breakdowns and project timelines.
6. Multiple Estimates: Users can receive and compare estimates from multiple constructors for a single request.
7. Notifications: Users and constructors should receive notifications for important actions (e.g., new estimates, report submissions).
8. **Solution Overview**

* **Three-Tier Architecture:** The solution is structured as a three-tier application with separate presentation, business logic, and data layers.
* **Microservices Architecture:** Each business function is managed by individual microservices, facilitating flexibility and scalability. The microservices communicate via REST APIs and are registered with a Eureka server for service discovery and load balancing.
* **Features:**
  + **User Interface:** Allows users to register, log in, create project requests, view estimates, and manage wishlists.
  + **Constructor Dashboard:** Enables constructors to view and manage project requests, submit cost estimates, and generate reports.
  + **RESTful API Communication:** The frontend will interact with backend services to manage data flow securely and efficiently.
* **Microservices:**
  + **User Management Service:** Manages user registration, authentication, profile management, and JWT-based authorization.
  + **Cost Estimation Service:** Calculates project costs based on user requests and material quality (basic, premium, moderate).
  + **Report Service:** Manages project lifecycle, tracking status (e.g., start, in-progress, completed).
  + **Constructor Service:** **:** Manages Constructor registration, authentication, profile management, and JWT-based authorization.
  + **Area Request Service:** Provides data visualizations and insights into cost trends and user-constructor interactions.

1. **Features and Functionalities**
   * **User Features**:
     + Create requests for area that is to be estimated.
     + Access the reports generated by the constructor.
     + View reports generated by different constructors.
   * **Constructor Features**:
     + View all the area requests generated by users.
     + Estimate the cost for user’s area request based on various factors.
     + Share the report to user
   * **Other Functionalities**:
     + JWT-based authentication for user and constructor.
     + Admin dashboard to view all the reports.
2. **Architecture Diagram**
3. **Technical Stack**

# **Presentation Layer (Frontend)**

* **Framework**: React with TypeScript
* **Libraries**: Material-UI , Bootstrap for UI components
* **Communication**: RESTful APIs via HTTP to interact with backend servicesThis layer will handle user interaction for contractors, homeowners, and designers, allowing them to manage projects, access cost estimates, input project data, and view real-time information.

# **Business Logic Layer (Backend)**

* **Framework**: Node.js with TypeScript (strictly)
* **Microservices Architecture**: Each microservice handles a distinct business function such as:
* **Eureka Server**: For service discovery and load balancing of microservices

# **Data Layer (Database)**

* **Database**: MongoDB (Non-relational, scalable) to store:
* **User Information**: Profiles of homeowners, contractors, and designers
* **Project Data**: Details of ongoing and completed projects
* **Material and Labour Prices**: Data for real-time cost estimation
* **Estimation History**: Records of previous project estimations
* **Caching**: Optional Redis or in-memory database for faster data retrieval (e.g., frequently requested estimates).

1. **Prerequisites and Requirements**

* **Development Environment Requirements**
* **Node.js (v14 or higher)**: For running the backend server and installing dependencies.
* **Git**: Version control to manage source code.
* **Visual Studio Code (VS Code)**: Recommended IDEs for coding in TypeScript

* **Project Setup and Management Tools**
* **GitHub** : For version control and collaboration.
* **Postman**: For API testing to ensure each microservice works independently and integrates correctly.

### **Development Prerequisites**

* **Strong Knowledge of TypeScript, and React.js**: For frontend development.
* **Experience with Node.js, Express, and RESTful API Development**: For backend service development.
* **Familiarity with MongoDB** : For data management.
* **Understanding of Microservices Architecture** : For effective development and deployment in a distributed environment.

1. **Future Improvements**

* Payment gateway integration for secure transactions.
* Real-time chat between users and constructors.
* Track constructors estimation history and recommend areas

1. **Consclusion:**

The **Construction Cost Estimation** project provides a streamlined solution to address the complexities and inaccuracies often encountered by homeowners and contractors when estimating construction or renovation costs. By implementing a robust three-tier, microservices-based architecture, this application allows users to generate detailed, comparative cost estimates based on project specifications. This approach enhances accuracy, optimizes the planning process, and provides financial and timeline projections that empower both homeowners and contractors.