

FREELANCE PROJECT BIDDING SYSTEM

~ Eshaan Bhardwaj (24BCS12637)

1) Project Overview

The Freelance Project Bidding System is a web-based platform designed to connect freelancers with employers through a transparent and competitive bidding mechanism. Employers can post projects with defined budgets and deadlines, while freelancers can browse available projects and submit bids including proposals, timelines, and cost estimates.

The system supports three distinct user roles- Freelancer, Employer, Administrator. Each role has dedicated dashboards and functionalities to ensure smooth project posting, bidding, hiring, and system monitoring.

2) Objectives

The main objectives of the Freelance Project Bidding System are:

- To provide a centralized platform for employers and freelancers to collaborate.
- To enable fair and transparent bidding on freelance projects.
- To simplify the project hiring and management process.
- To ensure secure authentication and role-based authorization.
- To provide an intuitive user interface for all stakeholders.
- To maintain scalability, reliability, and data integrity.

3) System Architecture

Presentation Layer (Frontend)

- Built using React.
- Provides separate portals for Freelancers, Employers, and Admins.
- Handles user interaction, form submission, and API consumption.

Application Layer (Backend)

- Developed using Spring Boot.
- Exposes RESTful APIs for authentication, project management, bidding, and hiring.
- Implements business logic and validation.
- Manages security and role-based access.

Data Layer (Database)

- PostgreSQL stores users, projects, bids, and transaction-related data.
- Uses relational constraints to ensure data consistency.

4) High-Level Design (HLD)

Main Modules

- Authentication & Authorization Module
- Employer Panel
- Freelancer Portal
- Bid Management System
- Admin Dashboard

High-Level Flow

- User registers and logs in.
- Role-based dashboard is displayed.
- Employers post projects.
- Freelancers place bids.
- Employers review bids and hire freelancers.
- Admin monitors users and system activities.

5) Low-Level Design (LLD)

Key Entities

User- userId, name, email, password, role

Project- projectId, title, description, budget, deadline, employerId, status

Bid- bidId, projectId, freelancerId, proposal, bidAmount, expectedTimeline, status

Core APIs- POST /auth/login, POST /auth/register, POST /projects, GET /projects, POST /bids, GET /bids/project/{id}, PUT /bids/{id}/accept

Frontend Components

Login/Register Pages, Employer Dashboard, Freelancer Dashboard, Project Listing Page, Bid Dashboard, Admin Panel

6) Tech Stack

Frontend- React.js

Backend- Spring Boot

Database- PostgreSQL

Security- JWT + Spring Security

7) Security Architecture

The system implements multiple security measures:

- JWT-based authentication for stateless sessions.
- Role-based authorization using Spring Security.
- Password encryption using BCrypt.
- Secure API endpoints with access restrictions.
- Input validation to prevent SQL injection and XSS.
- CORS configuration for controlled frontend-backend interaction.

8) Functional Requirements

Freelancer

- Register and login.

- View available projects.
- Submit bids with proposals and timelines.
- Track bid status.
- Manage profile information.

Employer

- Register and login.
- Post new projects.
- View bids for posted projects.
- Hire freelancers.
- Manage ongoing and completed projects.

Admin

- Manage users (activate/deactivate).
- Monitor projects and bids.
- Handle reported issues.
- Ensure platform integrity.

General

- Secure authentication.
- Responsive UI.
- Real-time data updates (optional enhancement).

9) Conclusion

The Freelance Project Bidding System successfully provides a scalable and secure solution for managing freelance projects and competitive bidding. By leveraging React, Spring Boot, and PostgreSQL, the system ensures high performance, modularity, and maintainability. The

role-based design improves usability for different stakeholders, while robust security mechanisms protect user data and system integrity.