



**UGANDA CHRISTIAN
UNIVERSITY**

A Centre of Excellence in the Heart of Africa

FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY DEPARTMENT OF COMPUTING AND
TECHNOLOGY

EASTER 2025 SEMESTER WEB PROGRAMMING PROJECT REPORT

PROGRAM: BSCS 2:2

COURSE: Web Programming

COURSE LECTURERS: Muganga Charles & Wambete Joseph

PROJECT TITLE: Online Job Posting and Application System

Submitted by: Group 8

Reg Number	Name
S23B23/047	Obba Mark Calvin
S23B23/020	Nicole Mary Johnson
S23B23/006	Arapta Hilda Lucky
M23B23/023	Leticia Lakica
S19B23/008	Bwengye Gilbert
S20B23/003	Kasongo Nathan

Date Submitted: 26th March 2025

Table of Contents

CHAPTER 1: ABSTRACT.....	3
Key Features:.....	3
CHAPTER 2: INTRODUCTION, PROBLEM STATEMENT, AND PROJECT OBJECTIVES	4
Introduction	4
Problem Statement.....	4
Project Objectives	4
The Application’s Structure.....	4
CHAPTER 3: METHODS, TOOLS, AND DESIGNS USED FOR THE PROJECT	8
Methods.....	8
Tools Used.....	8
Database Design.....	8
Database Tables and Relationships	8
Database Name: job_portal	8
Tables and Relationships	8
Relationships Summary	9
Key Functionalities Implemented	9
1. User Authentication & Role Management	9
2. Job Posting & Management	10
3. Job Search & Application System	10
4. Admin Controls.....	10
5. Notification System	10
6. Database Operations.....	10
7. Security Features	10
CHAPTER 4: CONCLUSION & RECOMMENDATIONS	11
Conclusion.....	11
Recommendations for Future Enhancements	11
1. Escrow Payment System.....	11
2. Skill Badge Verification	11
3. Video Profile Portfolios.....	11
4. AI-Powered Salary Estimator	11
5. Browser Extension	11
6. Integrated Video Interviews	12
7. Chat Sentiment Analysis	12
8. Gamification System.....	12

CHAPTER 1: ABSTRACT

The WorkFlow platform is a centralized online system designed to streamline job postings and applications, making the hiring process more efficient for employers and job seekers. The system allows employers to post jobs, track applications, and manage candidates, while job seekers can search, filter, and apply for jobs easily. Additionally, an admin role oversees the platform, managing users and job listings.

The platform follows modern web development standards, utilizing React.js for an interactive frontend, Node.js with Express.js for a scalable backend, and MySQL with Sequelize ORM for database management. The system implements secure authentication using JWT (JSON Web Tokens) and follows the microservices architecture to enhance modularity and maintainability.

Key Features:

- **Multi-Role System:** Secure authentication for Job Seekers, Employers, and Administrators with JWT-based access control.
- **Advanced Job Search:** Filters such as salary and skills.
- **Freelance & Full-Time Listings:** Support for gig-based work and traditional employment.
- **Employer Analytics Dashboard:** Track application rates, candidate demographics, and hiring trends.
- **It is built using React.js (Frontend), Node.js/Express.js (Backend), and MySQL (Database),** the platform ensures scalability, security, and performance.

CHAPTER 2: INTRODUCTION, PROBLEM STATEMENT, AND PROJECT OBJECTIVES

Introduction

The job market has become increasingly digital, with various platforms offering job listings. However, many of these platforms lack centralization, making it difficult for job seekers to find the right opportunities and for employers to reach the right candidates efficiently. The Work Flow system addresses this issue by providing a centralized and user-friendly job portal

Problem Statement

Employers are typically faced with the task of having to sort through large volumes of applications manually in an error-prone and time-consuming process, while job seekers are faced with poorly implemented search functionality, limited filters, and no centralized systems, so they have to create multiple accounts on different sites, input the same information repeatedly, and find it difficult to keep their applications organized, and hence it leads to inefficiency, frustration, and missed matches for both of them, and the necessity for a centralized, well-designed recruitment system that optimizes the process and makes the labor market more efficient and equitable is more apparent than ever. The Work Flow system solves these issues by integrating advanced search filters, real-time notifications, and a structured job application process.

Project Objectives

- Develop a centralized platform for job postings and applications.
- Implement secure user authentication for employers, job seekers, and administrators.
- Enable advanced search and filtering such as skills, location and job type.
- Provide real-time notifications for job status updates.
- Ensure data security and scalability through modern web technologies.

The Application's Structure

Capstone project/

```
|— frontend/
|   |— client/
|   |   |— public/
|   |       |— index.html
|   |       |— favicon.ico
|   |       |— manifest.json
|   |
|   |— src/
|       |— components/
|           |— Navbar.js
|           |— Footer.js
|           |— JobCard.js
|           |— JobSearch.js
|           |— EmployerDashboard.js
|           |— ApplicationList.js
|       |— pages/
|           |— Home.js
|           |— Login.js
|           |— Register.js
|           |— JobDetails.js
|           |— PostJob.js
```

```
| | | └─ Applications.js
| | | └─ Dashboard.js
| | └─ services/
| | | └─ authService.js
| | | └─ jobService.js
| | | └─ applicationService.js
| | └─ App.js
| | └─ index.js
└─ backend/
    └─ server/
        └─ controllers/
            └─ authController.js
            └─ jobController.js
            └─ applicationController.js
        └─ routes/
            └─ authRoutes.js
            └─ jobRoutes.js
            └─ applicationRoutes.js
        └─ models/
            └─ User.js
            └─ Job.js
            └─ Application.js
        └─ middleware/
            └─ auth.js
```

- | |— server.js
- | |— package.json
- | |— .env
- | |— README.md

CHAPTER 3: METHODS, TOOLS, AND DESIGNS USED FOR THE PROJECT

Methods

- Agile Development: We followed an iterative milestone-based approach, ensuring gradual improvements and testing with a focus on github for collaboration.
- Micro services and MVC Architecture: The code is in different files and separated the logic as well into models, views, and controllers for better maintainability.

Tools Used

- Frontend: React.js (UI), React Router (Navigation), Axios (API requests).
- Backend: Node.js with Express.js (API framework), Sequelize ORM (Database management).
- Database: MySQL since its a relational database system).
- Authentication: JWT (JSON Web Token) for secure login and user access.
- GitHub: For project tracking and collaboration.

Database Design

The database for WorkFlow is built using MySQL and managed with Sequelize ORM. It follows a relational structure with key entities such as Users, Jobs, and Applications.

Database Tables and Relationships

Database Name: job_portal

Tables and Relationships

Users Table (users)

- Stores job seekers, employers, and admins.
- **Key Fields:** id, email, password, role, first_name, last_name.

Company Profiles (companies)

- Stores employer company details.
- **Key Fields:** id, employer_id (FK), company_name, industry, website.

Job Seeker Profiles (job_seeker_profiles)

- Stores job seekers' qualifications.
- **Key Fields:** id, user_id (FK), title, skills, resume_url.

Jobs (jobs)

- Stores job listings posted by employers.
- **Key Fields:** id, company_id (FK), title, salary_range, job_type, status.
- **Indexes:** salary, experience_level, skills (FULLTEXT).

Applications (applications)

- Tracks job applications by seekers.
- **Key Fields:** id, job_id (FK), job_seeker_id (FK), status, cover_letter.

Relationships Summary

- **One-to-One:** Users ↔ Job Seeker Profiles, Users ↔ Companies.
- **One-to-Many:** Companies ↔ Jobs, Jobs ↔ Applications.
- **Many-to-Many:** Users ↔ Jobs (via Applications).

Key Functionalities Implemented

1. User Authentication & Role Management

- JWT-based secure login/registration

- Role-based access control (Job Seeker, Employer, Admin)
- Password hashing (bcrypt)

2. Job Posting & Management

- Employer dashboard for creating/editing job listings
- Job categorization (full-time, part-time, freelance)
- Application tracking system

3. Job Search & Application System

- Filtering by:
 - Location
 - Salary range
 - Job type
- One-click application submission
- Application status tracking (Pending/Reviewed/Rejected)

4. Admin Controls

- User management (suspend/delete accounts)
- Job listing moderation
- Analytics dashboard

5. Notification System

- Email alerts for:
 - New job matches
 - Application status updates
 - Employer messages

6. Database Operations

- MySQL relational database
- CRUD operations for all entities
- Indexed search queries

7. Security Features

- Input validation/sanitization
- Rate limiting on auth endpoints

- HTTPS encryption

CHAPTER 4: CONCLUSION & RECOMMENDATIONS

Conclusion

The WorkFlow job portal successfully provides a centralized, efficient, and secure system for employers and job seekers. By using modern web technologies such as React.js, Node.js, and MySQL, the platform ensures scalability, maintainability, and an enhanced user experience.

Recommendations for Future Enhancements

1. Escrow Payment System

- Implementation: Integrate Stripe/PayPal APIs
- Purpose: Securely hold freelance payments until job completion
- Benefit: Reduces payment disputes just like Upwork's escrow system

2. Skill Badge Verification

- Implementation: Admin-verified skill assessments
- Purpose: Validate candidate competencies which is similar to Upwork tests
- Benefit: Increases employer trust in applicant qualifications

3. Video Profile Portfolios

- Implementation: Allow video uploads/showcases
- Purpose: Candidates demonstrate work just like LinkedIn video posts
- Benefit: Enhances profile visibility beyond text resumes

4. AI-Powered Salary Estimator

- Implementation: Machine learning model analyzing market data
- Purpose: Suggest competitive salary ranges for job postings
- Benefit: Reduces pay disparity and negotiation friction

5. Browser Extension

- Implementation: Chrome/Firefox add-on
- Purpose: One-click job applications

- Benefit: Streamlines application process for seekers

6. Integrated Video Interviews

- Zoom/Google Meet API integration
- Enable scheduling and recording within the platform

7. Chat Sentiment Analysis

- NLP-based tone detection in employer-candidate messaging
- Flag potentially unprofessional communication

8. Gamification System

- Reward points for:
 - Profile completion
 - Skill assessments passed
 - Timely employer responses
- Redeemable profile boosts, inspired by Duolingo streaks