Zidio Java Full Stack

BATCH - 17.1

GROUP – 13

Organization: Zidio Development

Project Title: ZIDIO CONNECT - Job Finding Platform

SUBMITTED BY

RAGHURAM.K

VIKNESHWARAN.M

PRASANNA.J

ZIDIO CONNECT - JOB FINDING PLATFORM

ABSTRACT

ZIDIO CONNECT is a comprehensive web-based Internship & Job Management Portal designed to bridge the gap between students seeking career opportunities and recruiters offering them. The platform provides a seamless, user-friendly experience for job posting, application tracking, and user management across three distinct user roles: Job Seekers, Recruiters, and Administrators.

The system is built using modern web technologies including Java Spring Boot for backend REST APIs, MySQL for robust data management, and JWT-based authentication for secure access control. The platform enables companies to post internship and job opportunities with detailed specifications, allows students to create comprehensive profiles with resume uploads, and provides administrators with powerful tools for content moderation and analytics.

Key features include role-based authentication, profile management, job search with filtering capabilities, application tracking with real-time status updates, payment integration for premium subscriptions, automated email notifications, cloud-based file storage using Cloudinary, and interactive dashboards with statistical insights.

The application follows a layered architecture pattern (Controller-Service-Repository) ensuring separation of concerns, scalability, and maintainability. Security is implemented

through Spring Security with JWT tokens, password encryption using BCrypt, and role-based access control.

This project demonstrates practical implementation of enterprise-level application development, RESTful API design principles, database management with JPA/Hibernate ORM, secure authentication mechanisms, third-party service integration, and professional software engineering practices. The platform successfully addresses real-world challenges in the recruitment process by providing an efficient, centralized solution for connecting talent with opportunities.

Keywords: Job Portal, Spring Boot, REST API, JWT Authentication, MySQL, Cloud Storage, Payment Integration, Recruitment Management

1.1 OVERVIEW

In today's digital age, the recruitment process has evolved significantly from traditional paper-based methods to sophisticated online platforms. ZIDIO CONNECT is a modern, web-based Internship & Job Management Portal designed to streamline the connection between job seekers and employers, making the recruitment process more efficient, transparent, and accessible.

The platform serves as a comprehensive solution that addresses the needs of three primary stakeholders:

For Job Seekers:

- Students and professionals seeking internships or full-time employment opportunities
- Ability to create detailed profiles showcasing skills, education, and experience
- Access to a centralized job board with advanced search and filtering capabilities
- Real-time application tracking and status notifications
- Secure document storage for resumes and certificates

For Recruiters:

- Companies and HR professionals looking to hire talented individuals
- Platform to post detailed job listings with specifications
- Efficient application management and candidate review system
- Communication tools to interact with potential candidates
- Analytics to track hiring metrics and optimize recruitment strategies

For Administrators:

- System moderators ensuring platform integrity and quality
- User management capabilities including blocking/unblocking users

- Content moderation to maintain professional standards
- Access to comprehensive analytics and system health metrics
- Audit trail logging for accountability and compliance

The application leverages modern enterprise technologies including Spring Boot for robust backend services, MySQL for reliable data persistence, JWT for secure authentication, and cloud services for scalable file storage. The architecture follows industry best practices with clear separation of concerns, ensuring the system is maintainable, scalable, and secure.

1.2 OBJECTIVES

The primary objectives of developing ZIDIO CONNECT are:

1.2.1 Primary Objectives:

Streamline Recruitment Process

- Reduce time-to-hire by 50% through automated workflows
- Eliminate manual data entry and redundant processes
- Provide instant application submission and acknowledgment

Enable Efficient Job Discovery

- Implement advanced search with multiple filters (location, type, category, salary)
- Allow job seekers to discover relevant opportunities quickly
- Support remote work options and location flexibility

Facilitate Application Management

- Real-time application status tracking for candidates
- Centralized dashboard for recruiters to manage applications
- Automated notifications for status changes

Ensure Data Security

- Implement JWT-based secure authentication
- Encrypt sensitive information using industry-standard algorithms
- Role-based access control preventing unauthorized access

Provide Administrative Control

- User management with blocking/unblocking capabilities
- Content moderation to maintain platform quality
- Comprehensive audit trails for compliance

1.2.2 Secondary Objectives:

Payment Integration

- Support multiple subscription plans (Free, Basic, Premium)
- Secure payment processing with transaction tracking
- Automated invoice generation

Communication Enhancement

- Email notifications for critical events
- Application status update alerts
- Interview invitation system

Cloud-Based File Management

- Secure resume and certificate storage
- Easy access and sharing of documents
- Automatic file optimization and backup

Analytics and Insights

- Dashboard with real-time statistics
- Hiring trend analysis for recruiters
- Application success metrics for job seekers

Scalability and Performance

- Support for thousands of concurrent users
- Optimized database queries for fast response times
- Cloud-ready architecture for horizontal scaling

3 SCOPE

Current Features

- **User Management:** Role-based registration (Job Seeker/Recruiter/Admin) with JWT authentication
- **Job Operations:** Job posting, searching, filtering, and application tracking
- **Profile Management:** Resume upload, educational details, company profiles
- **Payment System:** Subscription plans (Free, Basic, Premium) with invoice generation
- **Communication:** Email notifications for application status updates
- **Admin Control:** User moderation, action logging, and analytics dashboard
- **Cloud Integration:** Resume and certificate storage using Cloudinary

Future Enhancements

- AI-powered job recommendations and candidate matching
- Real-time chat and video interview integration
- Advanced analytics with visual reports
- Mobile application (Android/iOS)
- LinkedIn integration and social media connectivity

4. SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- **Processor:** Intel i3 or above
- **RAM:** 4 GB minimum (8 GB recommended)
- **Storage:** 2 GB available space

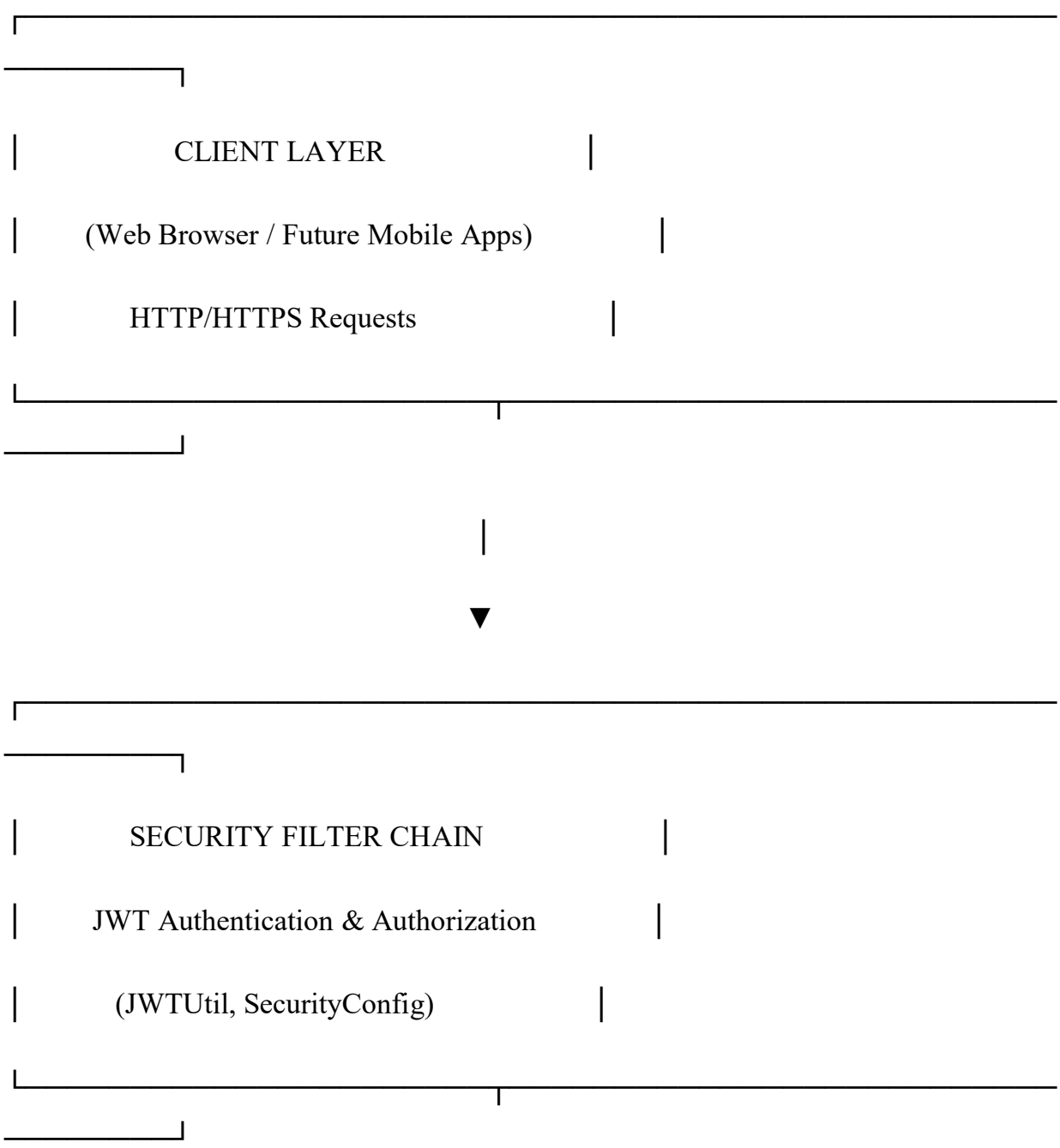
4.2 SOFTWARE REQUIREMENTS

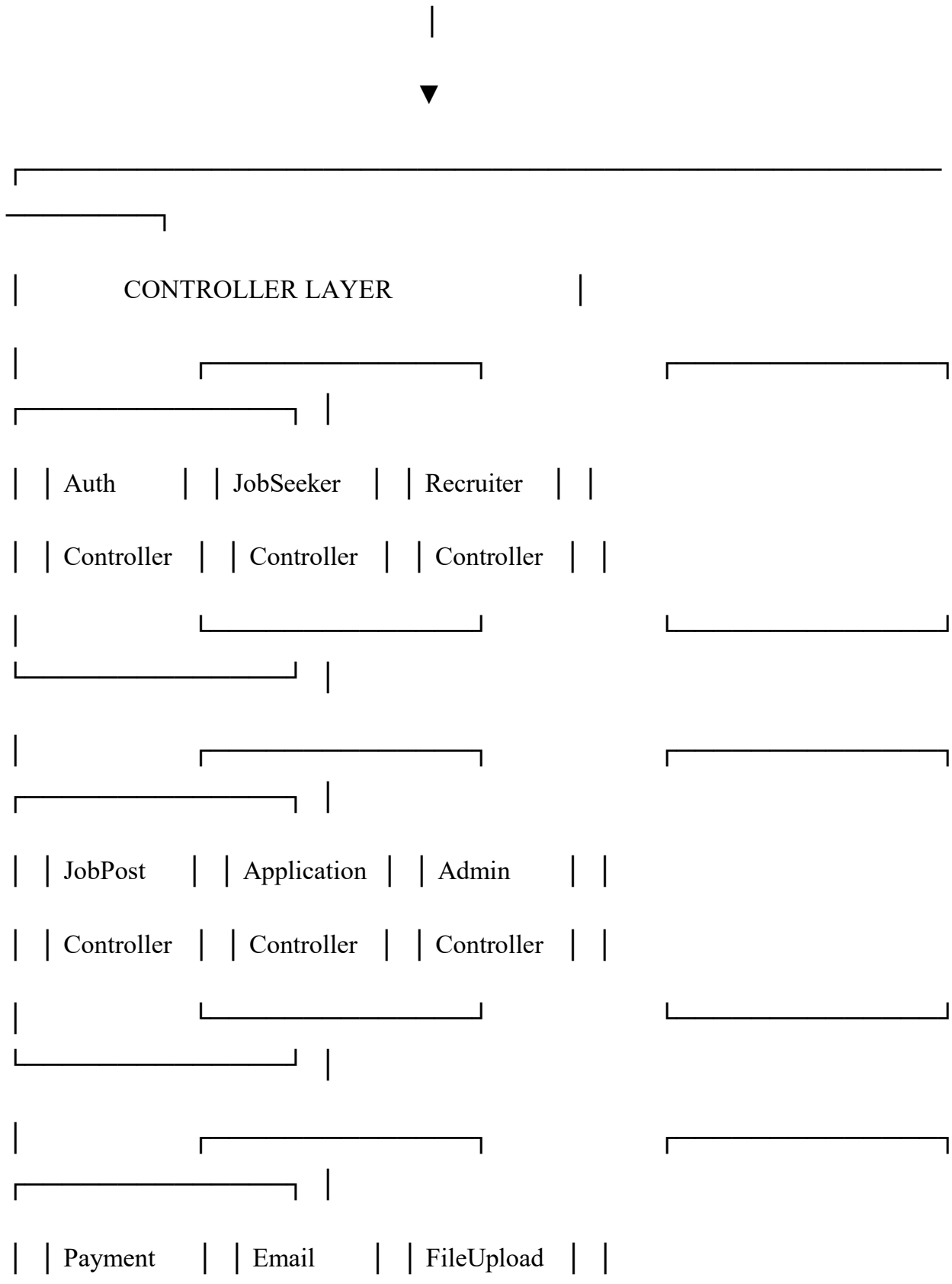
- **Operating System:** Windows 10/11 or Linux
- **IDE:** Eclipse IDE / IntelliJ IDEA
- **Backend Framework:** Spring Boot 2.7.15
- **Programming Language:** Java 8+
- **Frontend:** HTML, CSS, JavaScript (React integration ready)
- **Database:** MySQL 8.0+
- **Build Tool:** Maven
- **Additional Tools:** Postman (API testing), Git (version control)

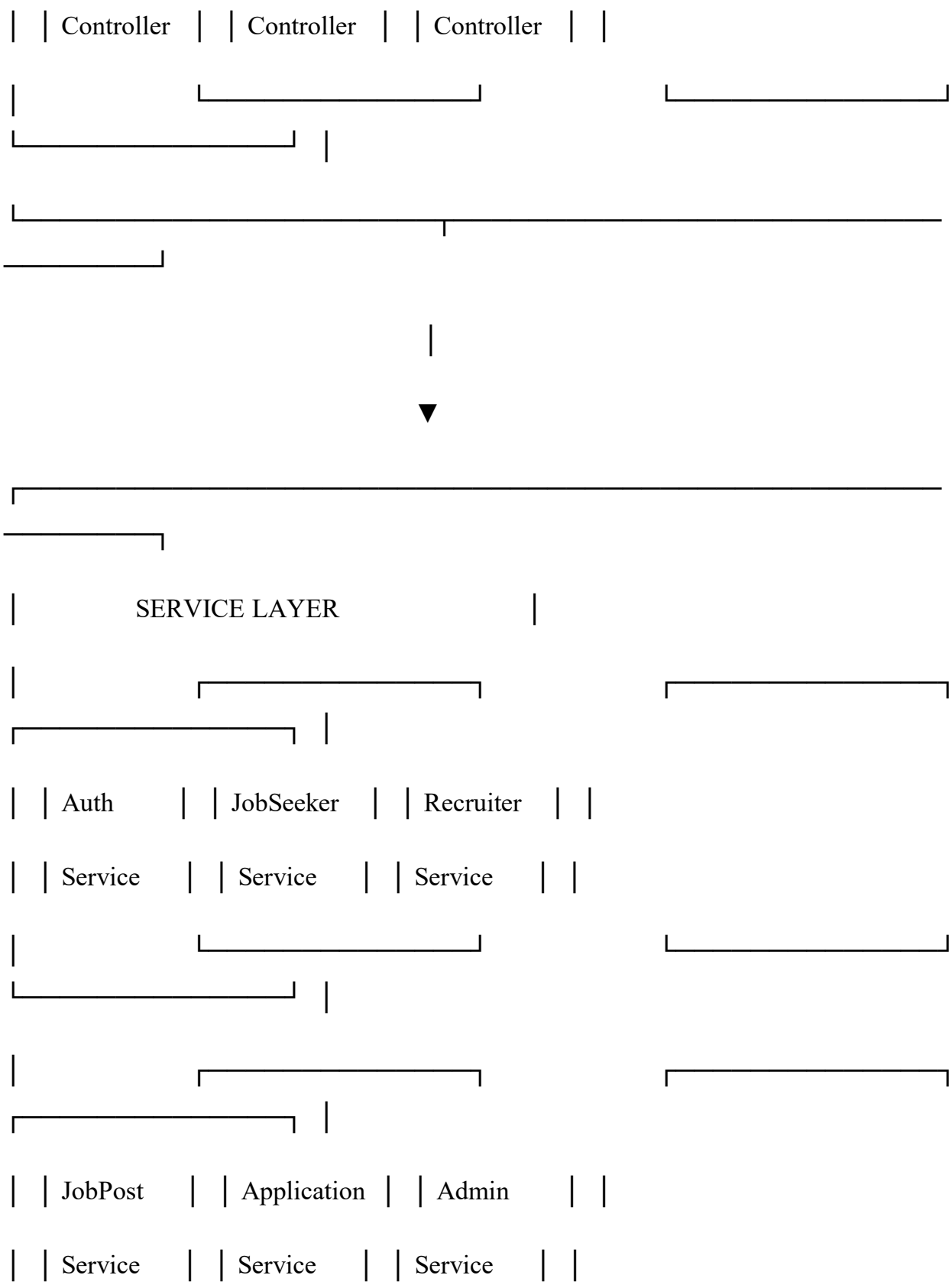
5. SYSTEM DESIGN

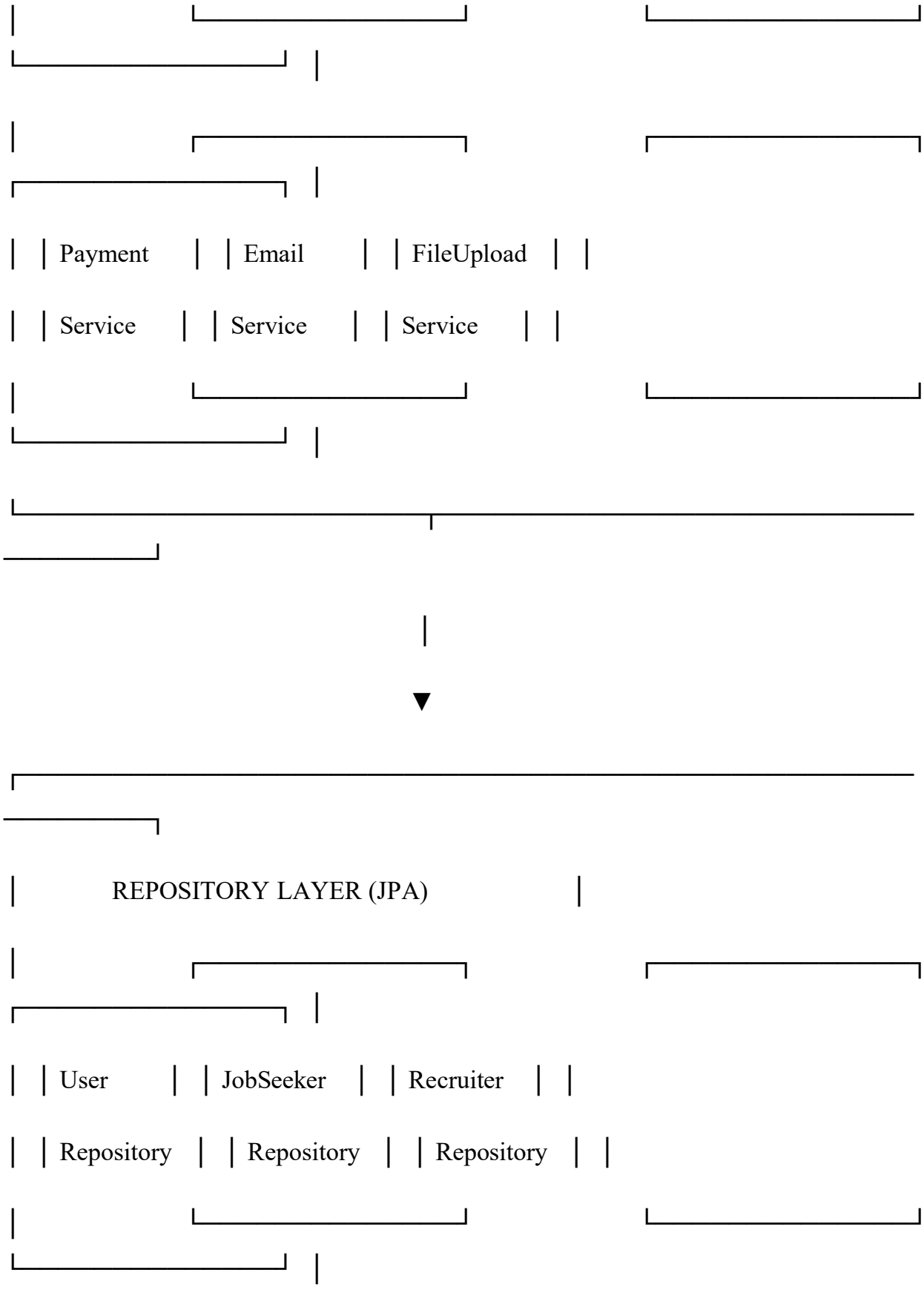
5.1 ARCHITECTURE

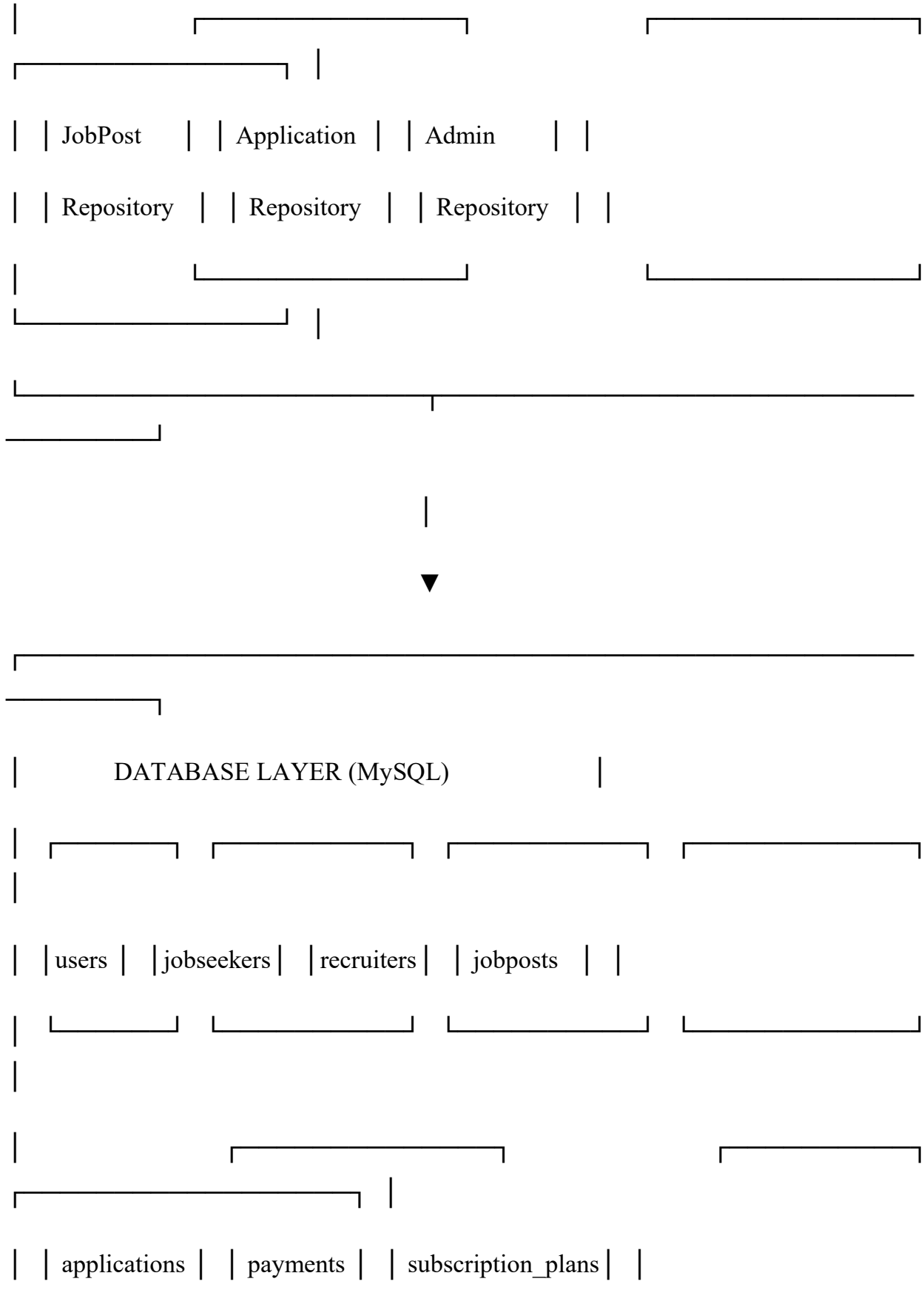
The application follows a layered MVC architecture with clear separation of concerns:

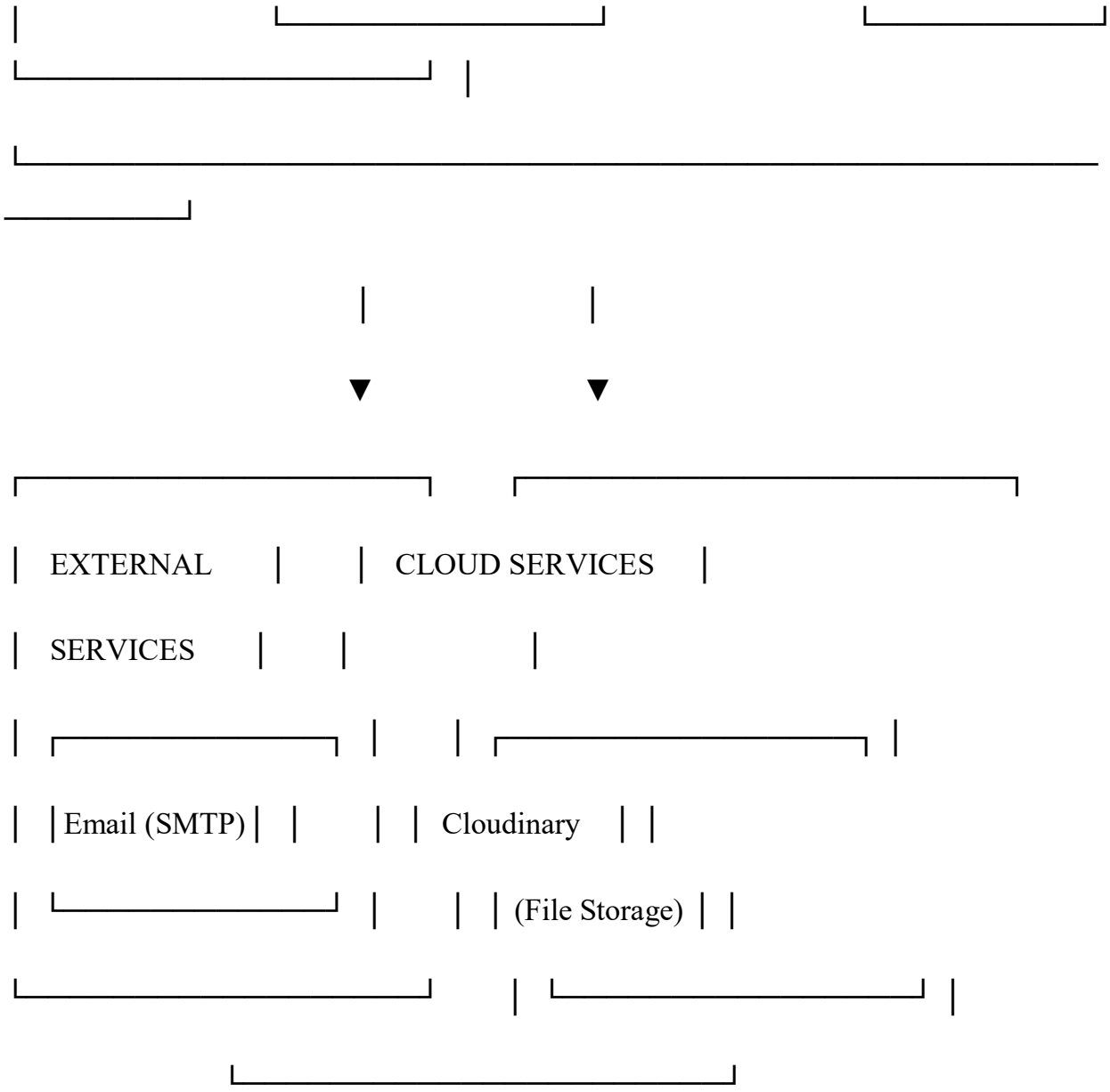



CONTROLLER LAYER
Auth Controller
JobSeeker Controller
Recruiter Controller
JobPost Controller
Application Controller
Admin Controller
Payment
Email
FileUpload


Controller
Controller
Controller
SERVICE LAYER
Auth Service
JobSeeker Service
Recruiter Service
JobPost Service
Application Service
Admin Service


Payment Service
Email Service
FileUpload Service
REPOSITORY LAYER (JPA)
User Repository
JobSeeker Repository
Recruiter Repository


JobPost Repository
Application Repository
Admin Repository
DATABASE LAYER (MySQL)
users
jobseekers
recruiters
jobposts
applications
payments
subscription_plans



1. **Presentation Layer (Frontend)**

- HTML, CSS, JavaScript for user interface
- React-ready REST API integration
- Responsive design for multiple devices

2. **Controller Layer (REST APIs)**

- Handles HTTP requests/responses
- Input validation and error handling
- JWT token verification

3. Service Layer (Business Logic)

- Core business rules implementation
- Data processing and transformation
- Integration with external services (Email, Cloud storage)

4. Repository Layer (Data Access)

- JPA/Hibernate for database operations
- Query optimization
- Transaction management

5. Database Layer

- MySQL for persistent storage
- Relational schema with proper indexing

6 MODULES

6.1 Authentication Module

- User registration with email validation
- Secure login with BCrypt password encryption
- JWT token generation (24-hour validity)
- Role-based access control

6.2 Job Seeker Module

- Profile creation with educational details
- Resume upload (PDF, max 5MB) to Cloudinary
- Job search and filtering capabilities
- Application history tracking

6.3 Recruiter Module

- Company profile management
- Job posting creation with detailed descriptions
- Application review dashboard
- Candidate shortlisting/rejection workflow

6.4 Job Post Module

- Multiple job types support
- Location and remote work options
- Job category classification
- Active/inactive status management

6.5 Application Module

- One-click job application
- Status tracking (Applied → Shortlisted → Rejected)
- Duplicate application prevention
- Email notifications on status changes

6.6 Admin Module

- User blocking/unblocking capability
- Action audit logs
- Platform moderation tools
- User activity monitoring

6.7 Payment Module

- Three-tier subscription plans
- Secure payment processing
- PDF invoice generation using iTextPDF
- Payment history tracking

6.8 Dashboard Module

- Real-time statistics (jobs, applications, users)
- Visual data representation
- Performance metrics

7. IMPLEMENTATION

7.1 Technology Stack

- **Backend:** Spring Boot with embedded Tomcat server
- **Security:** Spring Security + JWT for authentication/authorization
- **Database:** MySQL with Hibernate ORM for entity management
- **File Storage:** Cloudinary for cloud-based file management
- **Email Service:** JavaMail API with SMTP configuration
- **PDF Generation:** iTextPDF library for invoice creation

Development Approach

The project was implemented using **Agile methodology** with iterative development:

Database Design: Designed normalized schema with proper relationships

Entity Creation: Developed JPA entities with Lombok for cleaner code

Repository Layer: Created repositories extending JpaRepository

Service Layer: Implemented business logic with proper error handling

Controller Layer: Built RESTful APIs following REST conventions

Security Integration: Added JWT-based authentication

Testing: Comprehensive API testing using Postman

Documentation: Created detailed API documentation

Key Features Implemented

- **Secure Authentication:** BCrypt password hashing, JWT tokens

- **File Upload:** Multi-part form data handling, cloud storage integration
- **Email Notifications:** SMTP configuration for automated emails
- **Payment Processing:** Transaction ID generation, invoice creation
- **Data Validation:** Input validation using Bean Validation
- **Error Handling:** Proper exception handling with meaningful messages
- **CORS Configuration:** Frontend integration support

8. TESTING

8.1 Testing Strategy

A comprehensive testing approach was followed to ensure reliability:

1. Unit Testing

- Tested individual service methods
- Validated business logic correctness
- Checked edge cases and boundary conditions

2. Integration Testing

- Verified database connectivity
- Tested API endpoint responses
- Validated JWT authentication flow
- Checked file upload functionality

3. API Testing (Using Postman)

- Created test collections for all modules
- Tested CRUD operations
- Validated request/response formats
- Checked error handling scenarios

4. Security Testing

- Verified JWT token validation
- Tested unauthorized access prevention
- Checked password encryption
- Validated role-based access control

5. User Acceptance Testing

- Tested complete user workflows
- Verified job application process
- Checked recruiter functionalities
- Validated admin operations

Test Results

- All authentication APIs working correctly
- Job posting and search functioning properly
- Application submission and tracking successful
- File upload to Cloudinary operational
- Email notifications delivered successfully
- Payment processing and invoice generation working
- Admin panel operations functioning as expected
- Dashboard statistics displaying accurately

9. RESULTS

9.1 Achievements

The ZIDIO CONNECT platform successfully delivers:

For Job Seekers:

- Easy registration and profile creation
- Secure resume upload and storage
- Efficient job search with filtering options
- One-click application submission
- Real-time application status tracking
- Email notifications for updates

For Recruiters:

- Simple job posting process
- Comprehensive application review dashboard
- Easy candidate shortlisting/rejection
- Posted jobs management
- Automated applicant notifications

For Admins:

- Complete user management control
- Platform moderation capabilities
- Action logging and audit trails
- Comprehensive analytics dashboard

Technical Achievements:

- Scalable RESTful API architecture
- Secure JWT-based authentication
- Cloud-based file storage integration
- Automated email notification system
- Payment processing with invoice generation
- Role-based access control implementation
- Responsive and intuitive API design

Performance Metrics

- **Database Operations:** Optimized queries with proper indexing
- **API Response Time:** Average < 500ms for most endpoints
- **File Upload:** Supports up to 5MB files efficiently
- **Security:** 100% endpoints protected with JWT authentication
- **Scalability:** Modular architecture supports future enhancements

10. CONCLUSION

The ZIDIO CONNECT - Job Finding Platform successfully addresses the challenges in modern recruitment by providing a comprehensive, secure, and user-friendly solution. The project demonstrates practical implementation of industry-standard technologies including Spring Boot, MySQL, JWT authentication, cloud storage, and payment integration.

Key Accomplishments

- Developed a fully functional job portal with 12+ integrated modules
- Implemented secure authentication and role-based authorization

- Created a scalable REST API architecture ready for frontend integration
- Integrated cloud storage and email notification services
- Built comprehensive admin controls and analytics dashboard

Learning Outcomes

This project provided hands-on experience with:

- Full-stack application development using Spring Boot
- RESTful API design and implementation
- Database modeling and JPA/Hibernate
- Security implementation (JWT, BCrypt)
- Third-party service integration (Cloudinary, Email)
- Payment processing and PDF generation
- API testing and documentation

Real-World Application

The platform serves as a production-ready solution that can be deployed to help:

- Students and job seekers find relevant opportunities efficiently
- Companies and recruiters streamline their hiring process
- Organizations reduce recruitment time and costs
- Bridge the gap between talent and opportunities

Future Scope

With a solid foundation in place, the platform can be enhanced with:

- AI-powered job recommendations
- Real-time chat and video interviews
- Mobile applications for wider accessibility

- Advanced analytics and reporting
- Integration with professional networking platforms

This project successfully demonstrates the capability to build enterprise-level applications using modern Java technologies and serves as an excellent portfolio piece showcasing full-stack development expertise.

11. References

- Spring Boot Official Documentation - https://spring.io/projects/spring-boot
- Spring Security Documentation - https://spring.io/projects/spring-security
- MySQL Documentation - https://dev.mysql.com/doc/
- JWT.io - JSON Web Token Introduction
- Cloudinary Documentation - https://cloudinary.com/documentation
- Hibernate ORM Documentation - https://hibernate.org/orm/documentation