

Chapter 1: INTRODUCTION

1.1 Motivation

In recent years, Bangladesh has made remarkable progress in digital governance under the banner of "Digital Bangladesh." However, the pension management process for government employees remains largely manual, inefficient, and prone to delays or miscommunication.

This project, PensionProBD, aims to modernize and automate this system by developing a secure, role-based, dynamic pension management web application. The motivation behind this system is to ensure transparency, reduce corruption, and simplify the pension application process for both pension holders and government officers, with a modern UI that reflects Bangladeshi culture and language preferences.

1.2 Existing System

The current pension process in most government departments involves physical form submissions, multiple approval layers, and minimal tracking. Pensioners often face:

- Long waiting times for application processing.
- Lack of status transparency.
- Delayed response to complaints or document issues.
- Manual errors and data loss.

Officers, on the other hand, lack tools for quick verification, feedback sharing, or centralized document storage, making the overall system inefficient and error-prone.

1.3 Limitations of Existing System

Some key limitations of the traditional/manual system:

- No real-time tracking for applicants.
- High risk of human error in data handling.
- No secure document repository.
- Manual red flagging and misconduct tracking is almost non-existent.
- Difficult to retrieve past application data.

1.4 Problem Identification

The primary issues to be addressed are:

- Inability to track pension form status in real-time.
- No multi-role authentication or dashboard segregation.
- No formal complaint management system.
- Lack of responsive UI with native language (Bangla/English) toggle.
- Absence of secure and centralized document storage and downloadable reports.

1.5 Determining User Requirement

Based on interviews with current and retired government employees and discussions with government administrative officers, the system was designed to meet the following needs:

- Pension holders should submit forms only if they are eligible (minimum job_age = 19).
- Officers should be able to verify and comment on applications.
- Dashboard roles and access control must be clearly defined.
- Bangla/English language toggle.
- Real-time status notification system.
- Complaint system for accountability.

1.6 Summary of PensionProBD

PensionProBD is a role-based pension management system that allows:

- Pension Holders to register, submit pension forms, track status, upload documents, and raise complaints.
- Assistant Accountant General to verify, comment, and approve or forward applications.
- Head of Office to manage officers, resolve complaints, and monitor misconduct via red flag logic.

The system ensures transparency, multi-language support, cultural design, document security, and administrative control — making it a complete end-to-end digital solution for pension management.

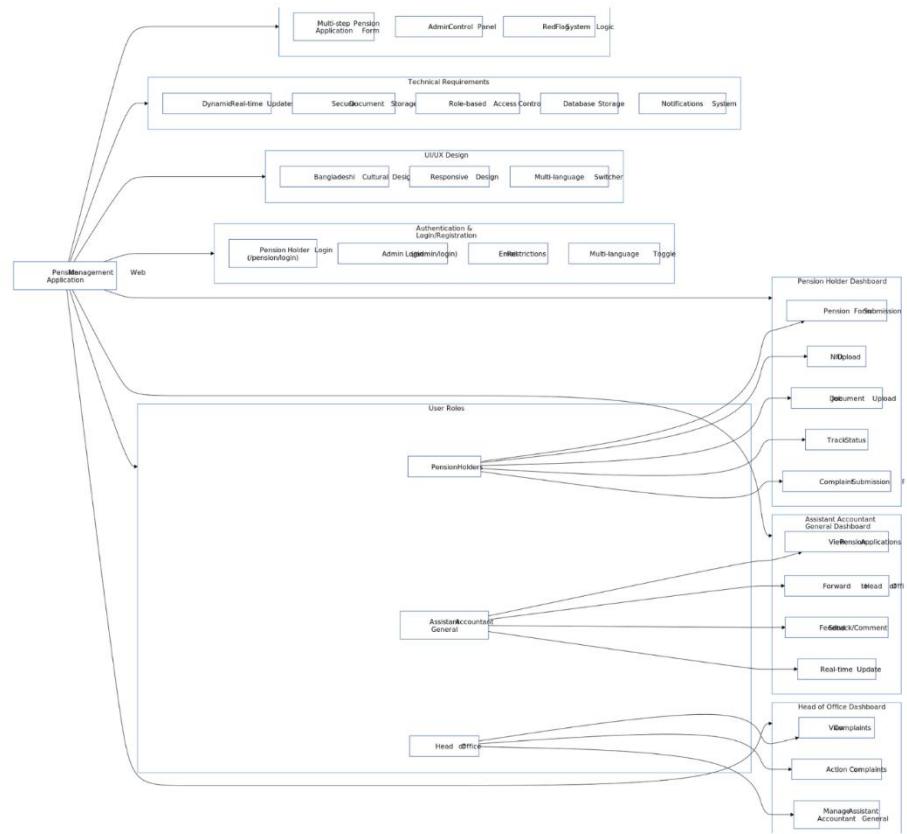


Diagram-01: System Overview Diagram

Chapter 2: LITERATURE REVIEW

2.1 Objective

The purpose of this chapter is to explore relevant research and existing systems related to pension management and role-based digital platforms. This review identifies the gap between existing pension systems and the need for a dynamic, secure, and accessible system tailored to Bangladeshi public sector employees.

2.2 Feasibility Study

A feasibility study was conducted to evaluate the practicality of developing PensionProBD. The study covered three major areas:

- **Technical Feasibility:** The proposed system will be developed using React.js for the frontend, Node.js for the backend, Tailwind CSS for UI, and MongoDB for data storage. These technologies are modern, scalable, and capable of supporting role-based access control (RBAC), file uploads, and real-time updates.
- **Operational Feasibility:** The system aims to streamline the pension application and approval process. It eliminates paperwork and simplifies communication between pensioners and officers, thus making it operationally viable.
- **Economic Feasibility:** As an academic project, no licensing or expensive tools are required. The open-source tech stack and hosting on platforms like Vercel or Render ensures minimal cost.

2.3 Facilities of PensionProBD

The proposed system introduces several features and improvements not typically present in older systems:

Feature	Benefit
Multi-role Dashboard	Individual dashboards for each user type with RBAC
Secure Document Upload	Prevents loss of hardcopy documents
Real-time Status Tracking	Improves trust and reduces confusion
Pension Form with Age Logic	Prevents premature or ineligible form submission
Red Flag Logic	Promotes accountability in officers
Bangla/English Toggle	Ensures accessibility for all user types
Notification System	Alerts for status changes or complaints
Admin Control Panel	Full user management and oversight by Head of Office

2.4 Summary

This literature review shows that while pension systems exist in other countries or sectors, most are either too complex, too manual, or not culturally adapted for Bangladesh. Systems like e-filing portals or general CRM platforms lack the dedicated structure needed for government pension workflows.

Thus, PensionProBD emerges as a solution combining the strengths of:

- Role-based authentication,
- Digital form management,
- Complaint and accountability tracking,
- And user-friendly UI tailored for the Bangladeshi context.

Chapter 3: SYSTEM ANALYSIS

3.1 System Development Life Cycle (SDLC)

The System Development Life Cycle (SDLC) provides a structured framework for developing software efficiently and effectively. For PensionProBD, the Agile Model was adopted to accommodate continuous feedback from users and iterative releases of working modules.

The SDLC phases followed are:

1. Requirement Analysis:

Identified user needs through interviews with pensioners and administrative officers.

2. System Design:

Created system architecture including data flow, class diagrams, and role-based dashboards.

3. Implementation:

Developed using MERN stack (MongoDB, Express.js, React.js, Node.js), focusing on RBAC and multi-language UI.

4. Testing:

Conducted unit, integration, and user acceptance testing for form submissions, role segregation, and red flag automation.

5. Deployment:

Hosted frontend on Vercel and backend on Render/Node environment.

6. Maintenance:

The architecture supports future updates like email alerts, CSV exports, and analytics.

3.2 Method Adopted

The project followed the Agile Development Methodology, which emphasizes:

- Incremental delivery
- Continuous user feedback
- Prioritization of user needs
- Adaptability to changing requirements

This method was ideal for our scenario, as different user roles (Pension Holder, Officer, Admin) needed customized workflows and interfaces that were refined with each sprint.

3.3 Basic Components of the System

The core system components of PensionProBD are:

➤ **Frontend Interface:**

- Built using React.js with Tailwind CSS.
- Responsive layout supporting desktop, tablet, and mobile views.
- Language toggle (Bangla/English).
- Conditional rendering of forms and features.

➤ **Backend Server:**

- Node.js with Express.js for API handling.
- JWT-based authentication for secure login.
- RBAC (Role-Based Access Control) logic.

➤ **Database:**

- MongoDB used for storing user data, pension applications, complaint logs, and red flag counts.
- Document uploads handled as file links stored securely.

➤ **Admin Control Panel:**

- Interface for Head of Office to manage users and complaints.

➤ **Notification System:**

- In-app real-time popup messages on form status changes, rejections, or approvals.

3.4 Definition of Software for the System

Here is the list of major software and tools used:

Software/Tool	Purpose
React.js	Frontend user interface
Tailwind CSS	Styling and UI layout
Node.js + Express.js	Backend server and API routing
MongoDB	NoSQL database for dynamic data handling
JWT	Secure token-based login authentication
Multer (Node Library)	File/document uploads
Vercel/Render	Hosting frontend/backend
Postman	API testing during development

Optional future integrations:

- EmailJS / Nodemailer for email alerts
- Chart.js / Recharts for dashboard analytics
- Google Translate API for language expansion beyond Bangla and English

3.5 System Analysis

After collecting all user requirements and reviewing the existing manual processes, the following analysis was conducted:

User Type	Required Features	Challenges Addressed
Pension Holder	Registration, NID/job doc upload, complaint form, PDF report	Confusion, long delays, no tracking
Assistant Accountant General	View, verify, comment, forward, reject applications	Manual errors, no documentation trace
Head of Office	Manage complaints, flag officers, manage accounts	Accountability, misconduct, bottlenecks

By implementing clear separation of roles and responsibilities along with digital tracking and cultural UI integration, PensionProBD transforms the legacy pension process into a modern, efficient solution.

Chapter 4: REQUIREMENT SPECIFICATION

4.1 Introduction

Requirement specification is the foundation of any software development process. This chapter outlines both functional and non-functional requirements for PensionProBD, as well as the technical environment, tools, and programming languages used during development.

The requirements were gathered based on:

- Interviews with pensioners and government officers,
- Existing pension form formats,
- Real-world workflow understanding of government offices.

4.2 User Requirements

Below are the primary user requirements, divided by role:

➤ Pension Holder

- Ability to register/login separately via a unique portal (/pension/login).
- Submit pension application form if $\text{job_age} \geq 19$ years.
- Upload NID and Job Certificate (PDF/Image).
- Track form status: Pending, Approved, Rejected.
- View officer comments or feedback.
- File complaints against officers.
- Download approved pension summary report (PDF).
- Switch site language between Bangla and English.

➤ Assistant Accountant General

- Common admin login (/admin/login).
- View and verify submitted pension forms.
- Add comments or corrections to applications.
- Forward verified forms to Head of Office.
- Reject incorrect submissions within 3 days.
- Receive complaints and notifications.

➤ **Head of Office**

- View all applications and complaints.
- Take action on complaints.
- Issue Red Flag warning if necessary.
- Auto-disable an officer account after 3 red flags.
- Add/edit/disable other officer accounts.
- Full control panel with dashboard insights.

4.3 System Requirements

➤ **Functional Requirements**

- Role-based login authentication (RBAC).
- Dynamic multi-step pension form.
- PDF report generation for approved forms.
- Document upload functionality (NID, Job Docs).
- Red flag logic for officer accountability.
- Multi-language UI (BD Bangla / us English).
- Admin control panel for user management.
- Complaint and response system.
- Real-time in-app popup notifications.

➤ **Non-functional Requirements**

- Secure data storage with access controls.
- Responsive design for mobile/tablet/desktop.
- Fast performance (load within 2s per route).
- Scalable backend to support growing users.
- Maintainability using modular codebase.

4.4 Programming Language

Layer	Language Used
Frontend	JavaScript (React.js)
Backend	JavaScript (Node.js with Express.js)
Styling	Tailwind CSS
Database Queries	MongoDB (Mongoose)

JavaScript was chosen across the full stack (MERN) to ensure efficient development and better maintainability.

4.5 Supporting Tools and Techniques

Tool/Technique	Purpose
Postman	API testing
JWT	Authentication token handling
Multer	File uploads
React Router	Route management
Tailwind CSS	Utility-first styling
MongoDB Atlas	Cloud database service
Dotenv	Secure environment variable management
React i18next	Language switching between Bangla/English

4.6 Summary

The requirement specification clearly outlines all technical and user-level needs of PensionProBD. The system is designed to support scalable, secure, and user-friendly pension management for the Government of Bangladesh. Adopting modern technologies ensures maintainability and performance, while the role-based access and UI personalization make it practical for daily use by all stakeholders.

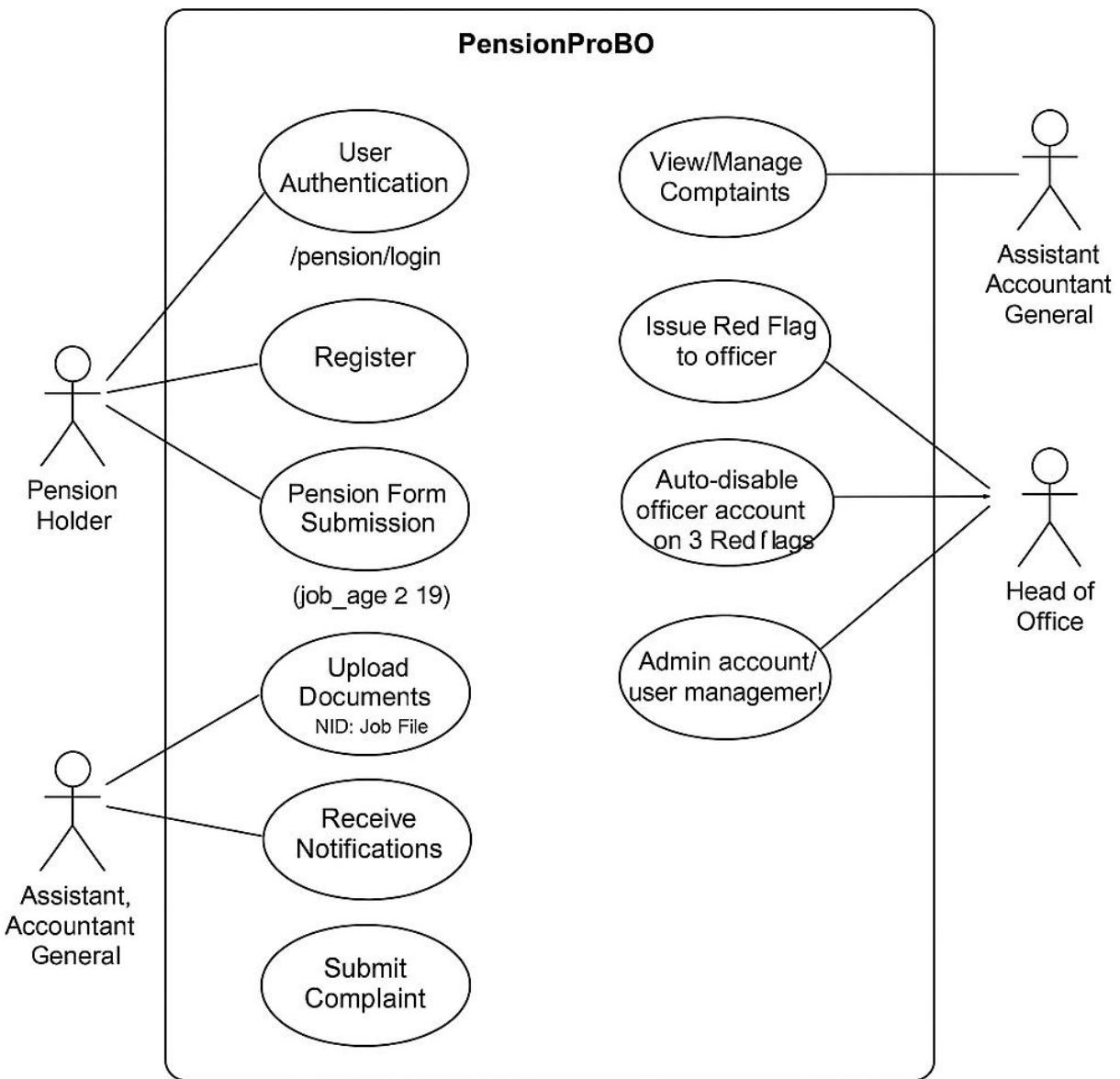


Diagram-02: Use Case Model

Chapter 5: SYSTEM DESIGN

System design is a critical phase that defines how software will be structured and how different components interact with each other. It includes architectural diagrams, flowcharts, and user interaction mappings that guide developers during implementation.

5.1 System Flow Chart

The System Flow Chart shows the logical sequence of operations and decisions for each user role within PensionProBD.

Diagram-03: System Flow Chart

5.2 Use Case Diagram

The Use Case Diagram demonstrates how users interact with the system, including the different types of users and their functionalities.

Actors:

- Pension Holder
- Assistant Accountant General
- Head of Office

Use Cases include:

- Register/Login
- Submit Pension Form
- Upload NID/Job Docs
- Review Applications
- Comment/Reject/Forward
- Manage Complaints
- Generate Pension Report
- Issue Red Flag

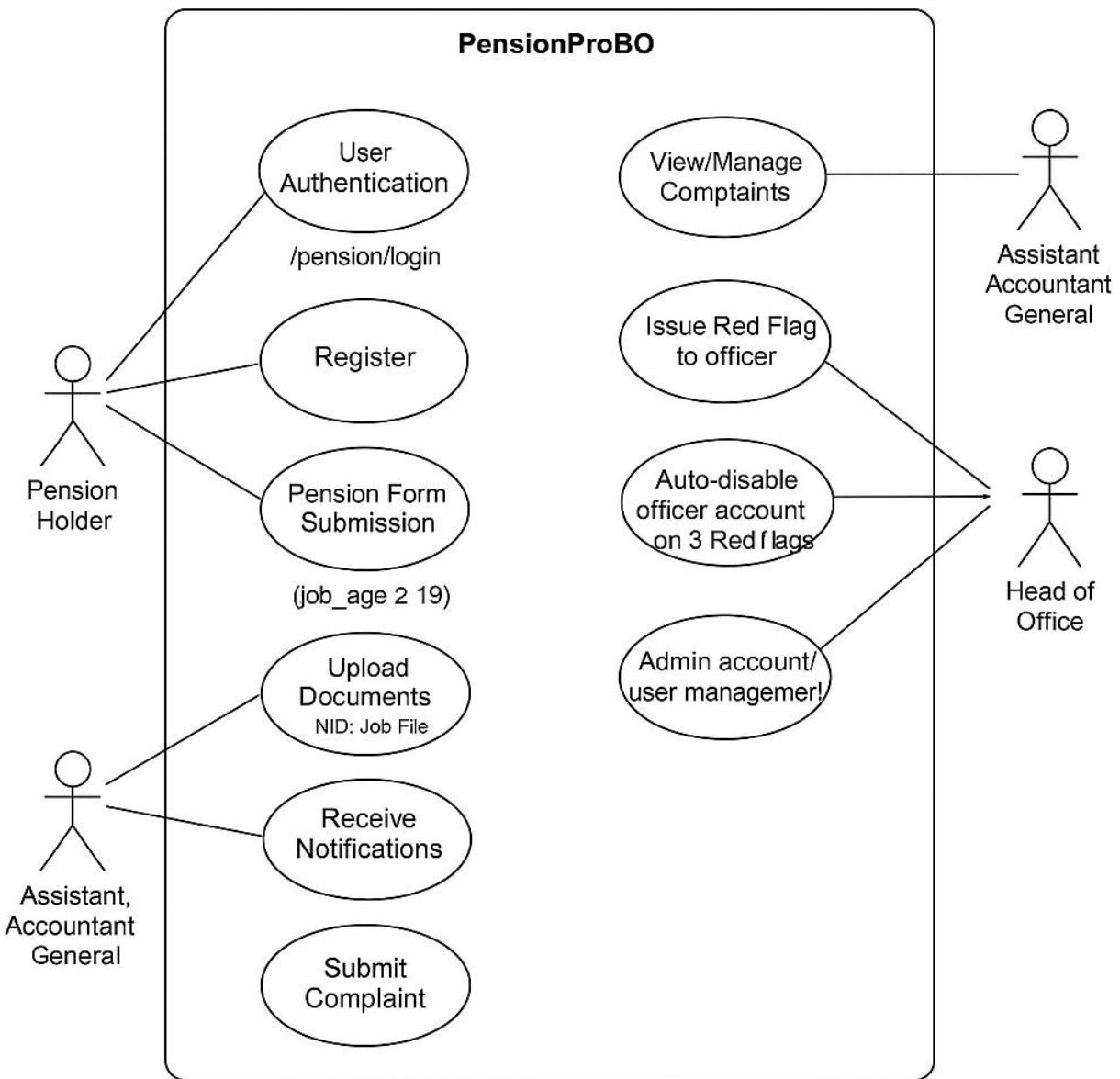


Diagram-04: Use Case Diagram

5.3 Data Flow Diagram (DFD)

- Level 0 (Context Diagram): Represents the entire system as a single process and how it interacts with external entities.

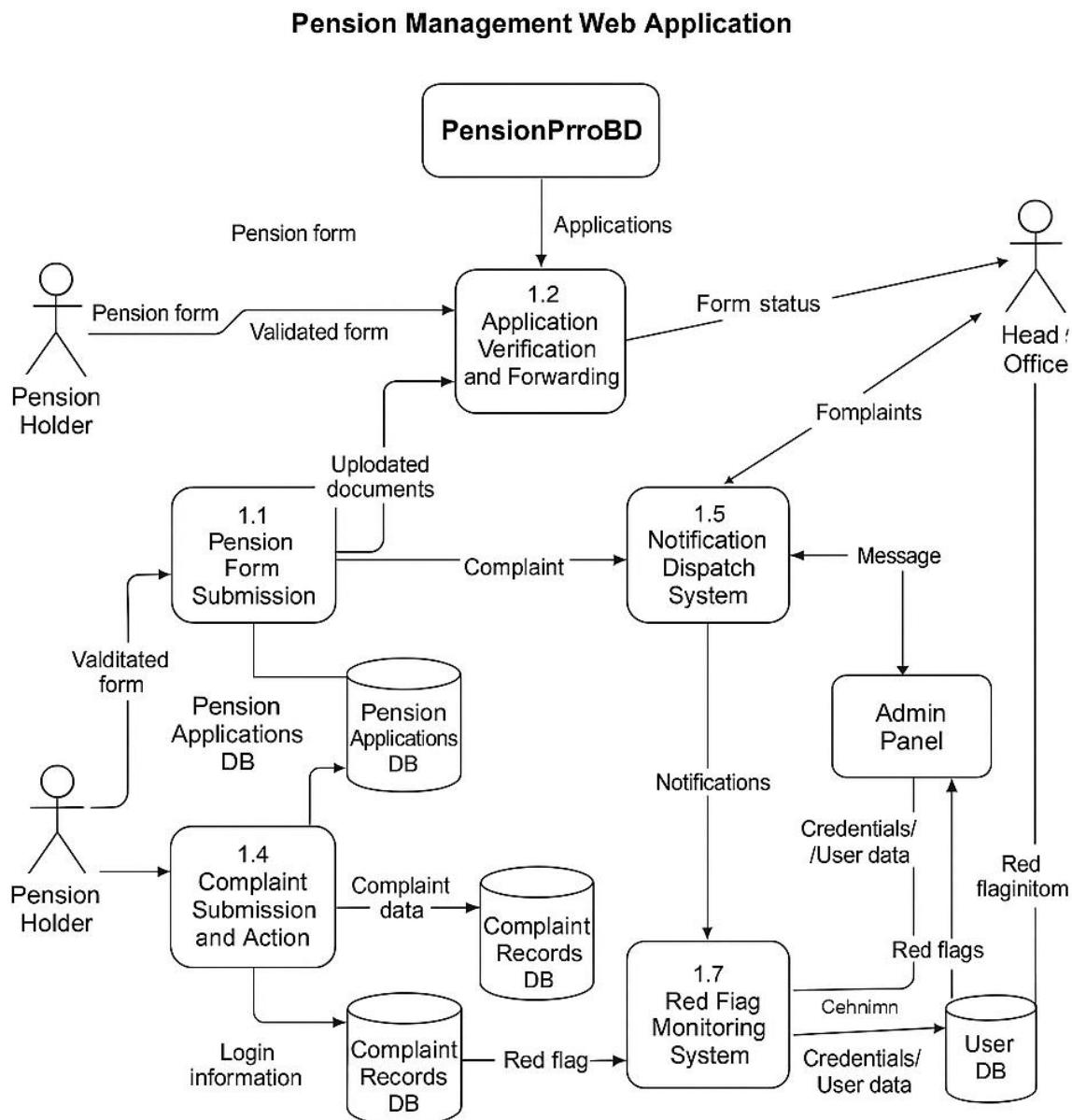


Diagram-05: Data Flow Diagram

➤ Level 1 (Process Breakdown):

Breaks the system into sub-processes such as:

- Form Submission
- Document Upload
- Verification
- Complaint Handling
- Admin Actions

5.4 Class Diagram

The Class Diagram represents the system's data model and the relationships between objects.

Key Classes:

- User (superclass with common fields)
- PensionHolder (subclass)
- AssistantAccountantGeneral (subclass)
- HeadOfOffice (subclass)
- PensionForm
- Document
- Complaint
- RedFlag

PensionProBD

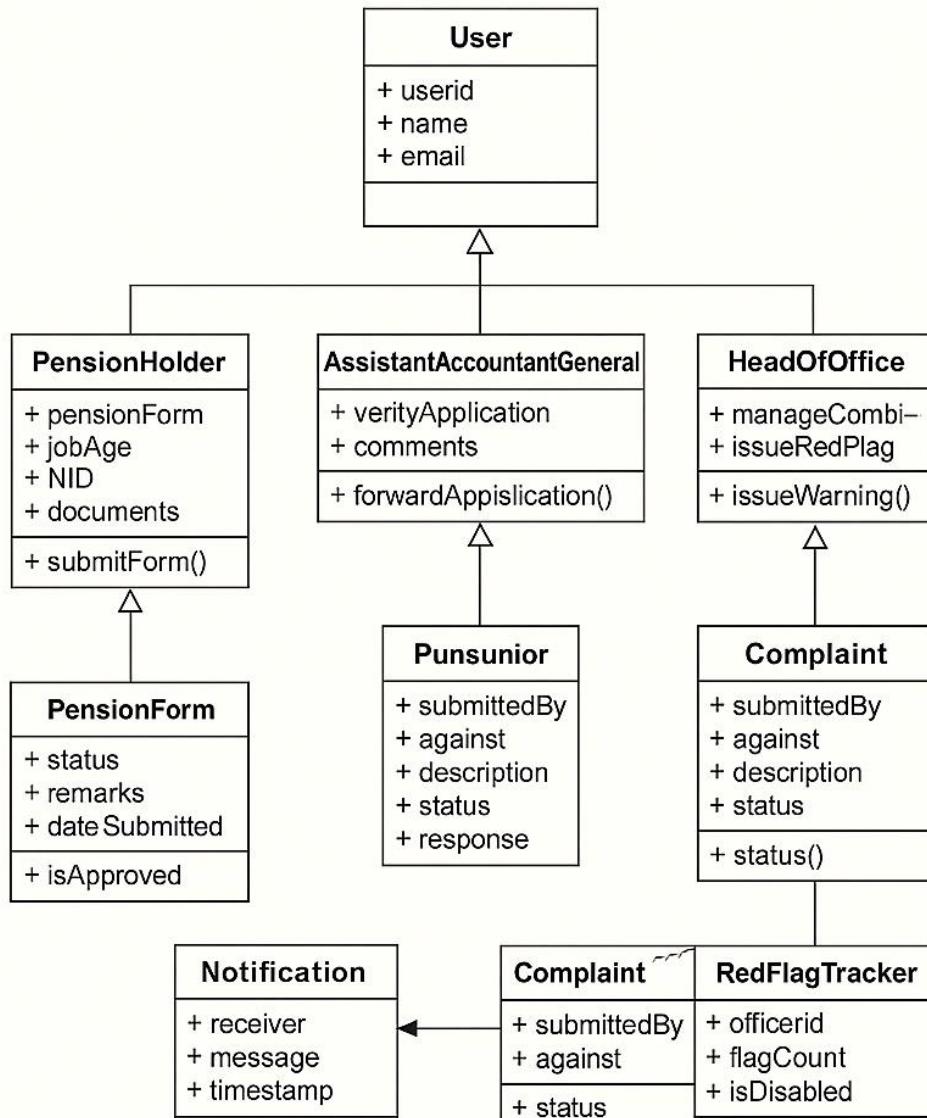


Diagram-06: Class Diagram

5.5 Entity Relationship Diagram (ERD)

The ERD shows the relationships among the major data entities:

- One Pension Holder submits one Pension Form.
- One Pension Form is reviewed by one Assistant Accountant General.
- Complaints can be raised by a Pension Holder against an officer.
- Head of Office oversees many officers and all complaints.

Diagram-07: Entity Relationship Diagram (ERD)

5.6 Site Map Design

A sitemap is used to describe the navigational structure of the application.

Key Routes:

- /pension/login – Login for Pension Holders
- /pension/register – Registration for new Pension Holders
- /admin/login – Login for Officers (AAG & Head of Office)
- /dashboard/user – Dashboard for Pension Holder
- /dashboard/aag – Dashboard for Assistant Accountant General
- /dashboard/admin – Dashboard for Head of Office
- /complaints, /form, /review, /pdf-download, etc.

5.7 UI Wireframe or Mockup

User Interface wireframes were created to design a clean, culturally inspired layout with

পেনশনপ্রোবিডি বিভিন্ন কার্যকরী পেনশন ব্যবস্থার সিস্টেম

বিশিষ্ট সম্পর্ক যোগাযোগ ও আপডেট সহ সহজে।

যোগাযোগ করুন।

ডিজিটাল পেনশন ব্যবস্থা পরিচালনা করুন।

আপনার আবেদন তার ক্রম। →

আপনারে সময়।

50K+
পেনশনদাতা।

95%
সফলতার হার।

24/7
সহায়তা।

কেন পেনশনপ্রোবিডি বেছে মেবেন?

আমাদের ব্যাপক ডিজিটাল প্লাটফর্মের সাথে পেনশন ব্যবস্থাপনা র ভবিষ্যত অনুভূত করুন।

Ahsan Ullah
pension_holder

আপনার পেনশন আবেদন করুন।

আবেদনের অবস্থা।

অভিযোগ জমা দিন।

আমার মাধ্যিক্র।

স্বাগতম, Ahsan Ullah!

Welcome to your pension management dashboard. Track your applications and manage your pension benefits.

Total Applications: 0

Pending Applications: 0

Approved Applications: 0

Total Complaints: 0

Recent Applications

No applications submitted yet.
Click on "Pension Application" to submit your first application.

Quick Actions

পেনশন আবেদন
Submit a new pension application

আবেদনের অবস্থা
Track your application status

অভিযোগ জমা দিন
Report any issues or concerns

Schedule Appointment
Book a meeting with an officer

BD পেনশনপ্রোগ্রাম বিভিন্ন ধরনের পেনশন যোগাযোগ সিস্টেম

Ahsan Ullah pension_holder

BD বাংলা ☀️ 🔔 Ahsan Ullah pension_holder ⏵

Personal Information

Full Name: Ahsan Ullah Father's Name:

Mother's Name: Date of Birth: mm/dd/yyyy

NID Number: Phone Number: 0175324512

Address: East Mollartek Udayan School Road Dhsaka-1230

পুরুষজী মহিলা

পরিষ্কার করো

প্রাপ্তির জন্ম তারিখ
পেনশন যোগাযোগ বিভাগ

BD পেনশনপ্রোগ্রাম বিভিন্ন ধরনের পেনশন যোগাযোগ সিস্টেম

A. Assistant Accountant assistant_accountant

BD বাংলা ☀️ 🔔 Assistant Accountant assistant_accountant ⏵

Welcome, Assistant Accountant
Assistant Accountant General Dashboard - Review and process pension applications

Pending Applications: 0

Reviewed Today: 0

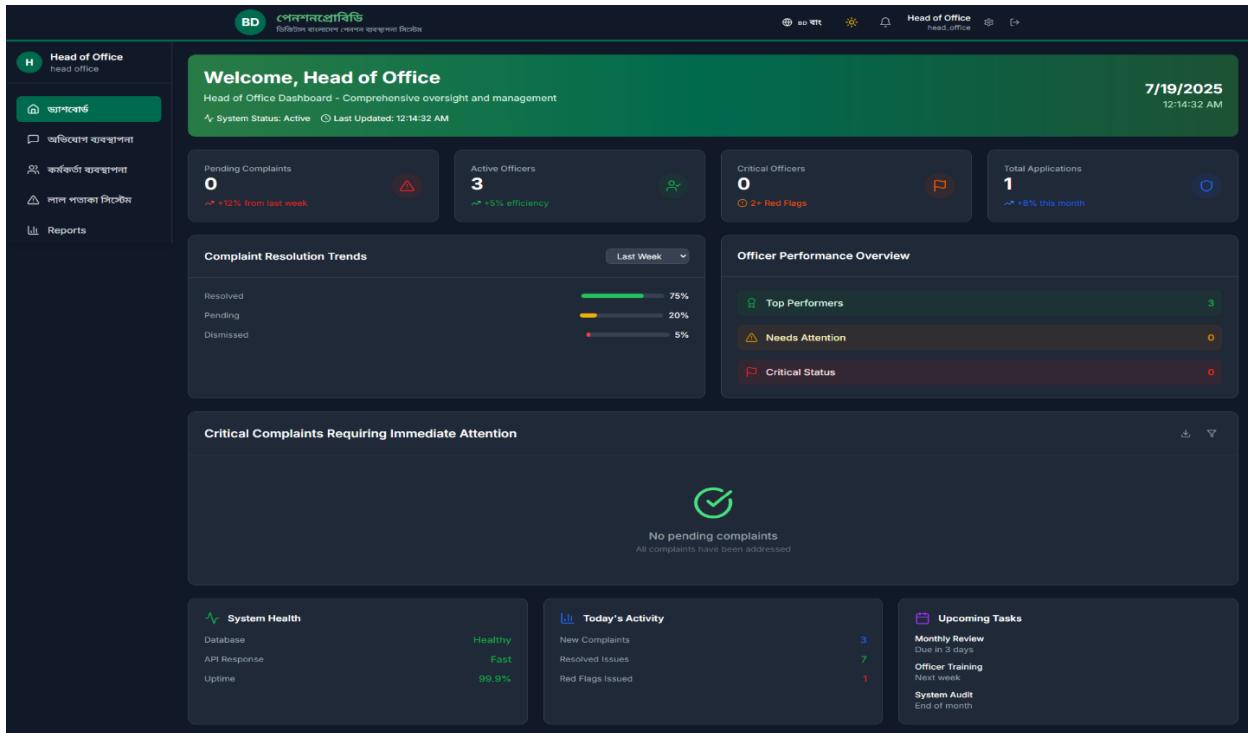
Total Processed: 1

Rejected: 0

Pending Applications

No pending applications

প্রাপ্তির জন্ম তারিখ
পেনশন যোগাযোগ বিভাগ



Wireframe Screenshots: Home Page, Dashboards, Form Submission Page, Admin Panel

5.8 Summary

This chapter outlines the architectural backbone of the PensionProBD system. Through flowcharts, diagrams, and models, it visualizes how users interact with the system and how data flows across different modules. The system is designed for scalability, cultural appropriateness, and practical use by various stakeholders involved in pension processing.

Chapter 6: SYSTEM IMPLEMENTATION AND TESTING

6.1 Introduction

This chapter discusses how the PensionProBD system was practically implemented using full stack technologies. It includes:

- Frontend and backend development strategies,
- Implementation of key features like form validation, document uploads, and multi-role dashboards,
- Followed by the testing methodology used to ensure functionality, reliability, and security.

6.2 Implementation Details

6.2.1 Frontend Implementation

The frontend was built using React.js with Tailwind CSS for styling and responsiveness. Key components include:

- Language Toggle: Using i18next to switch between BD Bangla and US English dynamically.
- Pension Form: A multi-step form that only becomes available if job_age >= 19.
- Login/Registration Pages: Role-specific routes for users and officers.

```
import { useTranslation } from 'react-i18next';

function LanguageToggle() {
  const { i18n } = useTranslation();

  const switchLanguage = (lang) => i18n.changeLanguage(lang);

  return (
    <div className="flex gap-2">
      <button onClick={() => switchLanguage('en')}>US English</button>
      <button onClick={() => switchLanguage('bn')}>BD বাংলা</button>
    </div>
  );
}
```

Language Toggle Component

6.2.2 Backend Implementation

The backend uses Node.js with Express.js, and JWT for authentication. Role-based logic ensures route access is restricted.

```
const authorizeRoles = (...roles) => {
  return (req, res, next) => {
    if (!roles.includes(req.user.role)) {
      return res.status(403).json({ message: 'Access denied' });
    }
    next();
  };
};
```

Role-Based Access Middleware

6.2.3 Document Upload Handling

File uploads (NID, job certificates) were handled using Multer, stored in a secure cloud directory.

```
const multer = require('multer');
const upload = multer({ dest: 'uploads/' });

router.post('/upload-docs', upload.fields([
  { name: 'nid' },
  { name: 'jobDocument' }
]), (req, res) => {
  res.send('Files uploaded successfully');
});
```

File Upload Route

6.2.4 Real-time Status Updates

The system uses React state updates and backend triggers to reflect real-time changes such as:

- Application forwarded to next level,
- Application rejected with comments,
- Complaint resolved.

6.3 Key Feature Implementation Summary

Feature	Technology
Language Switch	i18next (React)
Role Authentication	JWT + Middleware
Document Upload	Multer (Node.js)
Form Eligibility Check	React useEffect + Validation
Red Flag Logic	MongoDB update triggers
PDF Generation	React-to-PDF / jsPDF

6.4 Testing Methodology

Both manual and automated testing were conducted across user roles:

- Functional Testing
 - Verified form submission, routing, and file upload works.
 - Checked RBAC by logging in as different roles.
- Unit Testing
 - Used Jest for frontend components.
 - Verified form validation logic and role conditionals.
- Integration Testing
 - Ensured frontend and backend communicate properly (e.g., upload form -> store in DB -> notify next role).
- User Acceptance Testing (UAT)
 - Test users from mock pensioner and admin roles provided feedback.
 - Improvements made to button visibility, Bangla language accuracy, and form UX.

6.5 Summary

This chapter detailed the complete implementation of PensionProBD using modern full-stack technologies. Extensive testing ensured the system is stable, responsive, and secure. All key features—multi-role routing, complaint handling, red flag automation, and multi-language support—were successfully implemented and verified.

Chapter 7: RESULTS AND DISCUSSION

7.1 Introduction

This chapter highlights the results obtained from implementing the PensionProBD system, along with an in-depth discussion of its features, usability, limitations, and real-world value. It also reflects on how the system meets the initial requirements set during the analysis phase.

7.2 System Output

The project has successfully produced a fully functional pension management platform tailored to the Government of Bangladesh. Key outputs include:

Module	Output
Role-Based Login	Three types of secure logins with JWT: Pension Holder, Assistant Accountant General, and Head of Office
Pension Form	Dynamic, multi-step form that validates eligibility based on job age
Document Upload	Upload NID and Job Certificate securely
Real-time Status Updates	Pending → Verified → Approved → Rejected (with reason)
Complaint System	Allows pensioners to raise complaints and receive responses
PDF Summary	Pension amount report generated after final approval
Red Flag Logic	Auto-disables officers after 3 misconducts
Bangla/English Toggle	Full site can be switched between English and Bengali

7.3 Strengths of the System

- User-Friendly UI inspired by Bangladeshi government websites (green/red theme, formal typography).
- Secure login and access management.
- Fast performance through optimized routing and conditional rendering.
- Accountability system built in via red flags.
- Highly responsive for all screen sizes (mobile, tablet, desktop).

7.4 Limitations

Despite meeting the core objectives, the system has a few limitations:

- No built-in email notification system (optional future feature).
- No bulk CSV export or reporting tools yet.
- Red Flag system is fully logic-based, but lacks appeal process.
- No user analytics (e.g., how many users logged in today).
- Doesn't yet support OAuth login (Google/Facebook login).

7.5 Real-Life Use Case (Hypothetical)

Let's consider a user, Mr. Rahman, a government employee who is eligible for pension after 27 years of service.

1. Logs in to the PensionProBD portal.
2. Fills out the pension form, uploads NID and service certificate.
3. Sees live status updates: "Submitted" → "Verified" → "Approved".
4. Downloads final pension report in PDF.
5. If delayed, files a complaint. Officer receives a red flag if at fault.

This scenario confirms that the system mimics a real-life government workflow accurately.

7.6 Discussion

- The introduction of digital pension systems can drastically reduce paperwork, increase transparency, and minimize human error.
- Red Flag automation encourages better officer accountability without manual interference.
- Having multi-language support ensures accessibility to a broad audience, especially retired citizens.
- By using open-source and modern technologies, the platform is cost-effective and scalable for real deployment.

7.7 Summary

This chapter confirms that PensionProBD has successfully fulfilled its goals as a complete, culturally appropriate, and secure pension management solution for Bangladesh. The role-based structure and real-time tracking features solve long-standing issues in the existing manual system. Minor limitations exist but are easily solvable in future updates.

Chapter 8: CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The PensionProBD system successfully delivers a secure, role-based, and culturally adapted pension management web application tailored for the Government of Bangladesh. It replaces manual pension processing with a dynamic digital workflow involving three user roles—Pension Holder, Assistant Accountant General, and Head of Office—each with specific privileges and responsibilities.

The system ensures:

- Transparent pension form processing,
- Officer accountability through a red flag mechanism,
- Real-time feedback and status tracking,
- Complaint resolution management,
- A clean and accessible UI supporting both Bangla and English languages.

By using modern technologies like React.js, Node.js, MongoDB, and Tailwind CSS, the application is scalable, maintainable, and ready for further enhancements. The implementation demonstrates how Digital Bangladesh principles can be applied to real-life administrative problems.

8.2 Achievement Summary

- ✓ Multi-role authentication using JWT
- ✓ Form validation based on job age
- ✓ Secure document upload system
- ✓ Red Flag system for officer misconduct
- ✓ Real-time pension form tracking
- ✓ Downloadable pension summary in PDF
- ✓ Clean UI with national color palette and Bangla/English switch

8.3 Future Scope

Although the current system is functionally rich and production-ready, several improvements and extensions are possible for future deployment at scale:

1. Email & SMS Notifications
 - Notify pension holders and officers via email or mobile SMS.
 - Integration with Nodemailer or Twilio.

2. OAuth Login System
 - Google or Facebook login for faster access.
3. Officer Analytics Panel
 - Admin dashboard with insights like: number of forms approved/rejected, average response time, etc.
4. Appeal Mechanism for Red Flag
 - Allow officers to defend against wrongful complaints before auto-disabling.
5. PDF Archiving & Print-friendly Formats
 - Archive reports for 10+ years and add one-click “Print Version” for pension letters.
6. Integration with Government HR Database
 - Auto-verify employee service record and age from central database.
7. Mobile App Version
 - Lightweight Android/iOS app for pensioners who prefer mobile devices.

8.4 Final Thoughts

PensionProBD is more than just a student project—it's a prototype of how the digital transformation of government services can restore trust, increase speed, and simplify lives. If adopted by ministries or departments, this system could become a flagship model of digital governance for pensions in Bangladesh.

The project reflects the core values of:

- Digital inclusion,
- Transparency,
- Security, and
- Accessibility.

It has been a challenging yet rewarding experience that blends software engineering with social impact.

REFERENCES

Below is a list of sources, tools, and frameworks that have informed or supported the development of PensionProBD.

1. React.js Documentation – <https://react.dev/>
2. Node.js Documentation – <https://nodejs.org>
3. Tailwind CSS – <https://tailwindcss.com>
4. MongoDB Documentation – <https://www.mongodb.com/docs>
5. JWT Authentication Guide – <https://jwt.io/introduction>
6. Digital Bangladesh Vision 2021 – <https://a2i.gov.bd/digital-bangladesh>
7. Bangladesh National Portal – <https://bangladesh.gov.bd>
8. MERN Stack Tutorials – freeCodeCamp, YouTube, W3Schools
9. GitHub Open Source Projects (reference structure and design inspiration)
10. Gov.uk Design System – <https://design-system.service.gov.uk/>
11. PDF Make for client-side PDF generation – <https://pdfmake.github.io>
12. React Router DOM – <https://reactrouter.com>
13. Express.js Framework – <https://expressjs.com>
14. Formik & Yup for Form Validation – <https://formik.org> & <https://github.com/jquense/yup>
15. Tailwind UI Components – <https://tailwindui.com>
16. Sora UI Inspiration (Digital Bangladesh Branding) – <https://sora.gov.bd>
17. Bootstrap Bangladesh Themed Templates – Community & GitHub

APPENDIX

Appendix A: Technologies Used

- React.js – Powers the frontend as a Single Page Application (SPA).
- Node.js – Manages the backend server and core logic.
- Express.js – Handles API routing and middleware.
- MongoDB – Stores data using a flexible, document-based model.
- Tailwind CSS – Ensures responsive and utility-first UI design.
- JWT (JSON Web Tokens) – Secures user sessions with role-based access.
- PDFMake – Generates downloadable PDF summaries and reports.
- React Router – Handles client-side page navigation.
- Formik + Yup – Manages dynamic form state and validation.
- Cloudinary/Filebase – Optional solution for secure file and document uploads.

THE END