HIREX – Inclusive Job Portal

Shubham Bhowmick, Shubrajit Bagchi, Soumyadeep Mukherjee, Debojyoti Sen

Guide: Prof. Sampa Das

Siliguri Institute of Technology (2024–25)

Abstract & Objectives

- HIREX is an accessibility-first job portal designed to empower job seekers with disabilities.
- Fully adheres to Web Content Accessibility Guidelines (WCAG 2.1) standards.
- Helps users find jobs tailored to both their skills and accommodation needs.
- Provides resume builders, mock interviews, and career resources.
- Connects users with verified inclusive

Key Features

- HIREX includes features designed specifically for inclusivity and accessibility:
- Smart profiles highlight skills, preferred communication, and access needs.
- Job filters like 'Remote-friendly', 'ASL support', and 'Wheelchair-accessible'.
- Toolkits help employers understand inclusive hiring practices.
- Virtual job fairs with live captioning and interpreters.

Technology Stack

- We utilized modern and open-source technologies for full-stack development:
- Frontend: HTML, CSS, and JavaScript for clean, responsive UI.
- Backend: Node.js with Express.js for API logic and routing.
- Database: MongoDB with Mongoose for schema-based user data.
- Security: Bcrypt for password encryption, JWT for session control.

System Design & Architecture

- The project follows a modular and layered architecture:
- Backend organized into routes, models, and config for maintainability.
- Frontend is cleanly separated with HTML pages and JS API handlers.
- RESTful APIs manage interaction between frontend and backend.
- MongoDB provides flexibility in handling diverse user data.

Database & Security Measures

- Security and data consistency were core priorities:
- Mongoose schemas define strict data validation and uniqueness.
- Passwords are hashed using Bcrypt to prevent leaks.
- Sensitive information stored in environment variables.
- JWT handles secure logins and protected routes.

Authentication & User Flow

- A smooth authentication flow enhances user experience and security:
- Login/Signup handled by backend route `auth.routes.js`.
- Post-login, users are redirected to dashboards based on roles.
- Role-based views ensure employers and users see appropriate content.
- Frontend handles form validation and displays relevant error messages.

Testing & Debugging

- We ensured robust testing and quality assurance throughout:
- Unit testing of backend APIs to ensure reliability.
- Integration testing validates frontend-backend communication.
- Manual testing with Postman and browser DevTools.
- Test cases covered invalid input, duplicate signup, role mismatch.

Timeline & Cost Estimation

- A structured plan helped balance efficiency and cost-effectiveness:
- Backend completed in 2 weeks, frontend in 2 weeks, testing in 1 week.
- Total developer hours: approximately 100 (team effort).
- Used free-tier hosting tools like Vercel and MongoDB Atlas.
- Estimated developer value at $800–$1200.
- PERT and Gantt charts helped visualize

Conclusion & Future Scope

- The final implementation is solid and sets the stage for future improvements:
- Successfully developed a secure, role-based inclusive job portal.
- All core features are modular and scalable.
- Future upgrades: Use React for UI, add automated testing (Jest).
- Implement features like real-time messaging and password recovery.
- Live deployment and feedback loop will help

Thank You

Questions & Feedback Welcome!