DEI Builders – Systems Manual

1. Introduction

DEI Builders is a web-based career development platform designed to support college students and underrepresented talent in exploring career opportunities. The system provides tools such as a student dashboard, resume analyzer (demo-based), job matching (demo-based), mentorship requests, and workshops. This Systems Manual describes how to install, configure, and understand the technical architecture of the DEI Builders application.

The goal of this manual is to allow future student developers, instructors, or technical reviewers to understand how the system works, how to run it locally, and how to troubleshoot common issues.

2. Installation and Setup Instructions

2.1 Prerequisites

Before installing DEI Builders, ensure the following software is installed on your system:

- Node.js (version 18 or higher recommended)
- npm (comes with Node.js)
- Git
- A modern web browser (Chrome, Firefox, or Edge)

2.2 Project Setup

1. Clone the project repository from GitHub:

```
git clone <repository-url>
```

1. Navigate into the project directory:

```
cd deibuilders
```

1. Install project dependencies:

```
npm install
```

1. Start the development server:

```
npm start
```

1. Open a browser and navigate to:

```
http://localhost:3000
```

The application should load the DEI Builders home page.

---

3. Configuration Details

3.1 Environment Variables

The current version of DEI Builders is designed to function without required environment variables. Demo logic is used in place of live APIs to ensure consistent behavior during evaluation.

Optional (future use):

- `REACT_APP_GEMINI_API_KEY` – for AI-powered resume parsing
- `REACT_APP_JOBMATCH_API_KEY` – for external job matching services

These variables can be placed in a `.env` file at the project root if live APIs are enabled in the future.

3.2 Dependencies

Key dependencies used in this project include:

- React – Frontend framework
- react-router-dom – Client-side routing
- Jest & React Testing Library – Automated testing
- pdfjs-dist / mammoth – Resume file handling (PDF/DOC/DOCX support)

All dependencies are managed through `package.json` and installed via npm.

3.3 Database Setup

This version of DEI Builders does **not** use a live database. All data (resume results, job matches, dashboard status) is handled in memory using demo data structures. This approach was intentionally chosen to simplify deployment and ensure predictable results for academic evaluation.

---

4. System Architecture Overview

DEI Builders follows a modular, component-based architecture using React.

4.1 High-Level Architecture

- **Frontend (React SPA)**
- Home
- Login / Register
- Dashboard
- Resume Analyzer (Demo)
- Job Matches (Demo)
- Mentorship Requests
- Workshops
- **Routing Layer**
- Managed using `react-router-dom`
- **Logic Layer**
- Resume parsing demo pipeline
- Job matching demo pipeline
- **Testing Layer**
- Unit and UI tests using Jest
- Continuous Integration via GitHub Actions

The system is entirely client-side and runs in the browser.

---

5. API Documentation (Demo-Based)

5.1 Resume Analyzer (Demo)

Purpose: Simulates resume parsing results

Input:

- Resume file (PDF, DOC, or DOCX)

Output (Demo):

- Name
- Email

- Key Skills
- Education Summary
- Experience Summary

The demo logic guarantees output regardless of the uploaded file to demonstrate intended system behavior.

5.2 Job Matching (Demo)

Purpose: Displays job opportunities based on parsed resume data

Output (Demo):

- Job title
- Company name
- Location
- Required skills
- Match score (simulated)

Job matches are deterministic and designed to reflect the resume analyzer results.

---

6. Troubleshooting Guidance

Issue: Application does not start

- Ensure Node.js and npm are installed correctly
- Run `npm install` again to reinstall dependencies

Issue: Blank screen on load

- Check browser console for routing errors
- Verify `App.js` routes are correctly defined

Issue: Resume Analyzer shows demo data

- This is expected behavior in the current academic version

Issue: ESLint warnings appear

- Warnings such as `no-unused-vars` do not break the application
- These were left intentionally to prioritize functionality and clarity

---

7. Conclusion

This Systems Manual documents the technical setup and structure of the DEI Builders platform. The system was designed and implemented by a group of college students with a focus on clarity, usability, and academic demonstration. While some features are demo-based, the architecture supports future expansion into fully integrated AI and database-driven solutions.

DEI Builders demonstrates a complete, functional web system suitable for instructional evaluation and future development.