Software Engineer Assignment: Job Board Application

1. Introduction

The objective of this project is to build a full-stack job board application that crawls job postings from platforms like Naukri.com or LinkedIn Jobs and displays them in an interactive UI. The application will provide job listings for a specific keyword, with key attributes such as job title, company name, location, experience, and application link.

2. Project Requirements

2.1 Data Crawling

- Fetch job listings for a given keyword (e.g., "Product Manager").
- Extract key attributes: Job Title, Company, Location, Experience, and Application Link.
- Store job data in a structured format.

2.2 Backend Development

- Implement APIs to serve the crawled job data.
- Store job data in a database (MongoDB/PostgreSQL).
- Endpoints:
- GET /jobs - Fetch all job listings with optional filters.
- POST /refresh - Trigger job data update.
- Automate data refresh every 24 hours using a scheduled task.

2.3 Frontend Development

- Build a responsive UI with React.js or Next.js.
- Display job listings with filtering options (location, experience, etc.).
- Implement a search bar for job title queries.
- Create a job detail page with an "Apply" button.

2.4 Deployment

- Deploy the backend on Render/Heroku.
- Deploy the frontend on Vercel/Netlify.
- Store the database on MongoDB Atlas/PostgreSQL on Supabase.

3. Implementation

3.1 Tech Stack

- Frontend: React.js, Tailwind CSS
- Backend: Flask/FastAPI (Python) or Express.js (Node.js)
- Database: MongoDB/PostgreSQL
- Scraping: BeautifulSoup, Scrapy (Python) or Puppeteer (Node.js)
- Deployment: Vercel, Render, MongoDB Atlas

3.2 Database Schema


```
{
"id": 1,
"title": "Software Engineer",
"company": "Google",
"location": "Bangalore",
"experience": "3-5 years",
"apply_link": "https://example.com"
}
```


3.3 API Endpoints

- GET /jobs: Fetch job listings.
- POST /refresh: Refresh job data.
- GET /job/:id: Fetch specific job details.

4. Bonus Features

- Pagination or infinite scrolling.
- Automated job data update every 24 hours.

- Clean and minimalistic UI.

- Dockerized application for easy deployment.

5. Deliverables

- GitHub Repository (Code and Documentation).

- Live Deployed Application Link.

- PDF Report (This Document).

6. Conclusion

This job board application provides a seamless experience for users to find job listings efficiently. The automated data crawling and user-friendly interface enhance accessibility and usability. The system is scalable and can be expanded with more job sources and additional filtering options in the future.