**Course2Career**

**1. Introduction:**

The project is a full-stack job searching and posting website where the companies can post jobs related to particular domains. It serves as a platform for job seekers to explore jobs posted by companies and job posters to list open positions connected to educational courses.

**2. What?**

The website will allow users to:

* Job Posters can post jobs categorized by their courses.
* Job Seekers can view jobs categorized by their domains.

**3. Why?**

This platform bridges the gap between academia and industry, allowing universities to showcase job opportunities tied to specific fields of study, while companies can recruit talent based on their domains. It will make it easier for job seekers to find relevant opportunities and job posters to find suitable candidates.

**4. List of Supported Features:**

* Job posting by companies with domain categorization.
* Job seekers can view job postings by course or domain.
* Basic user authentication for job seekers and posters.
* Search functionality for job seekers to filter jobs based on courses/domains.

**5. List of Not Supported Features:**

* No support for internships at the moment.
* No advanced job recommendations based on user profiles.
* No automated email notifications for job seekers.

**6. List of Future Planned Features:**

* Internship postings by both universities and companies.
* Job recommendation engine based on user interests and skills.
* A notification system for job seekers (email alerts).
* Resume upload and download functionality.

**7. How**

**1. High-Level Diagram:**

* Diagram would include users (job seekers and posters) interacting with frontend (React/HTML/CSS), backend (Node.js/Express.js), and database (MongoDB).

Job seeker

Job poster

Frontend (HTML,CSS,React.js)

Backend (Node.js or Express.js)

Database (MongoDB)

**2. List of Components/Modules:**

* Frontend: Job posting page, Job seeker page, University/course/job listing page, Company/domain/job listing page.
* Backend: User Authentication module, Job posting API, University/Company module, Course/Domain module, Job listing API.
* Database: MongoDB collections for Universities, Courses, Companies, Domains, Job Postings, Users.

**3. Languages to be Used for Each Module:**

* Frontend: JavaScript (React), HTML, CSS.
* Backend: Node.js with Express.js
* Database: MongoDB.

**4. List of 3rd Party/Open Source Modules:**

* **Frontend:**
  + React.js, Axios for API calls.
* **Backend:**
  + Express.js for routing.
  + Mongoose for MongoDB interaction.
* **Authentication:**
  + Passport.js or JWT for user authentication.

**4.1 Table of Licenses:**

| **Module** | **License** |
| --- | --- |
| React | MIT |
| Express | MIT |
| MongoDB | Server Side Public License v1 (SSPL) |
| Passport.js | MIT |
| Axios | MIT |

**5. List of Any 3rd Party Services/APIs:**

* **Free:**
  + MongoDB Atlas (for database hosting).
  + AWS (for possible future file uploads or cloud functions).

**6. REST API Endpoints with Payloads:**

* POST /universities: Create a new university.

json

{ "name": "University A", "location": "Location A" }

* POST /universities/:id/courses: Create a new course for a university.

json

{ "name": "Computer Science", "department": "Engineering" }

* POST /companies: Create a new company.

json

{ "name": "Company A", "industry": "Software" }

* POST /companies/:id/domains: Create a new domain for a company.

json

{ "name": "Web Development" }

* GET /jobs: Retrieve a list of all jobs.
* GET /universities/:id/courses/:courseId/jobs: Get jobs for a specific course.
* GET /companies/:id/domains/:domainId/jobs: Get jobs for a specific domain.

**7. Build Steps/Scripts:**

* Install project dependencies: npm install
* Start development server: npm run dev

**8. Install Steps/Scripts:**

* Clone the project repository:

git clone <repository-url>

cd project-directory

* Install dependencies:

npm install

* Setup environment variables:
  + Create a .env file with MongoDB connection string and other necessary environment variables.
* Run the application:

npm run start

**8. GitHub Information:**

* The project repository can be hosted on GitHub. For Week 3, create a branch called week3 from the develop branch.
* Add the setup files and initial backend foundation to this branch.

**9. References:**

* MongoDB Documentation: <https://www.mongodb.com/docs/>
* Express.js Documentation: <https://expressjs.com/>
* React.js Documentation: <https://reactjs.org/>
* Passport.js Documentation: <http://www.passportjs.org/>