Project Name: Talent Hub - A Portal for Referral

Business Goal: The goal of Talent Hub is to streamline the candidate selection process for companies while providing candidates with personalized recommendations and learning opportunities to improve their skill sets.

Overall System: Talent Hub is a web-based application portal that facilitates the registration of candidates, allows them to input their details manually or upload resumes, matches them with suitable job openings based on their profiles and skill sets, conducts initial testing rounds, and provides personalized learning recommendations for candidates who fail to clear the testing rounds.

System Components:

1. **User Authentication and Access Control:** Manage user authentication and authorization to ensure secure access to the portal for candidates and authorized company personnel.
2. **Candidate Registration and Profile Management:** Enable candidates to register, input their personal details, and manage their profiles.
3. **Resume Parsing:** Implement a component to parse resumes uploaded by candidates and extract relevant information.
4. **Job Matching Algorithm:** Develop an algorithm to match candidates with job openings based on their profiles and skill sets.
5. **Testing Module:** Design a module to conduct initial testing rounds for candidates.
6. **Learning Recommendation Engine:** Implement an AI-based recommendation engine to suggest learning programs or courses for candidates based on their areas of improvement.
7. **Database Management:** Manage a centralized database to store candidate information, job openings, test results, and learning recommendations securely.
8. **Scalability and Performance:** Implement measures to ensure the system can scale up or down based on demand and maintain optimal performance.

Sub-Components:

1. **Frontend:** User interface for candidate registration, profile management, job search, and test-taking.
2. **Backend:** Logic for job matching, testing, learning recommendation, and database management.
3. **AI Module:** Machine learning models for resume parsing, job matching, and learning recommendations.
4. **Database:** Centralized storage for candidate and job data.

5. **Security Module:** Ensure secure access to the portal and data encryption to protect sensitive information.

AI-Based Prediction Modeling Scenarios:

1. **Resume Parsing:** Use natural language processing (NLP) to extract key information from resumes, such as skills, experience, and education.
2. **Job Matching:** Employ machine learning algorithms to match candidate profiles with job requirements based on skill similarity and experience.
3. **Learning Recommendations:** Utilize collaborative filtering or content-based filtering to recommend learning programs or courses tailored to the candidate's skill gaps.

Possible Use Cases:

1. Candidate Registration and Profile Creation
2. Resume Uploading and Parsing
3. Job Search and Matching
4. Testing Rounds
5. Learning Recommendations

Possible Limitations:

1. **Accuracy of Resume Parsing:** The accuracy of resume parsing may vary depending on the complexity and formatting of resumes.
2. **Job Matching Accuracy:** The accuracy of job matching depends on the effectiveness of the algorithm and the quality of candidate profiles and job descriptions.
3. **Learning Recommendations:** Recommendations may not always perfectly align with the candidate's learning needs and preferences.

Preferred Technology Stack (Open Source, Windows Environment):

- **Frontend:** React.js
- **Backend:** Node.js with Express.js
- **Database:** PostgreSQL
- **AI/ML:** Python with libraries like TensorFlow or PyTorch for modeling, NLTK or spaCy for NLP tasks
- **Security:** JSON Web Tokens (JWT) for authentication, HTTPS for secure communication
- **Scalability:** Docker for containerization, Kubernetes for orchestration

Tentative Timeline: (9-10 months)

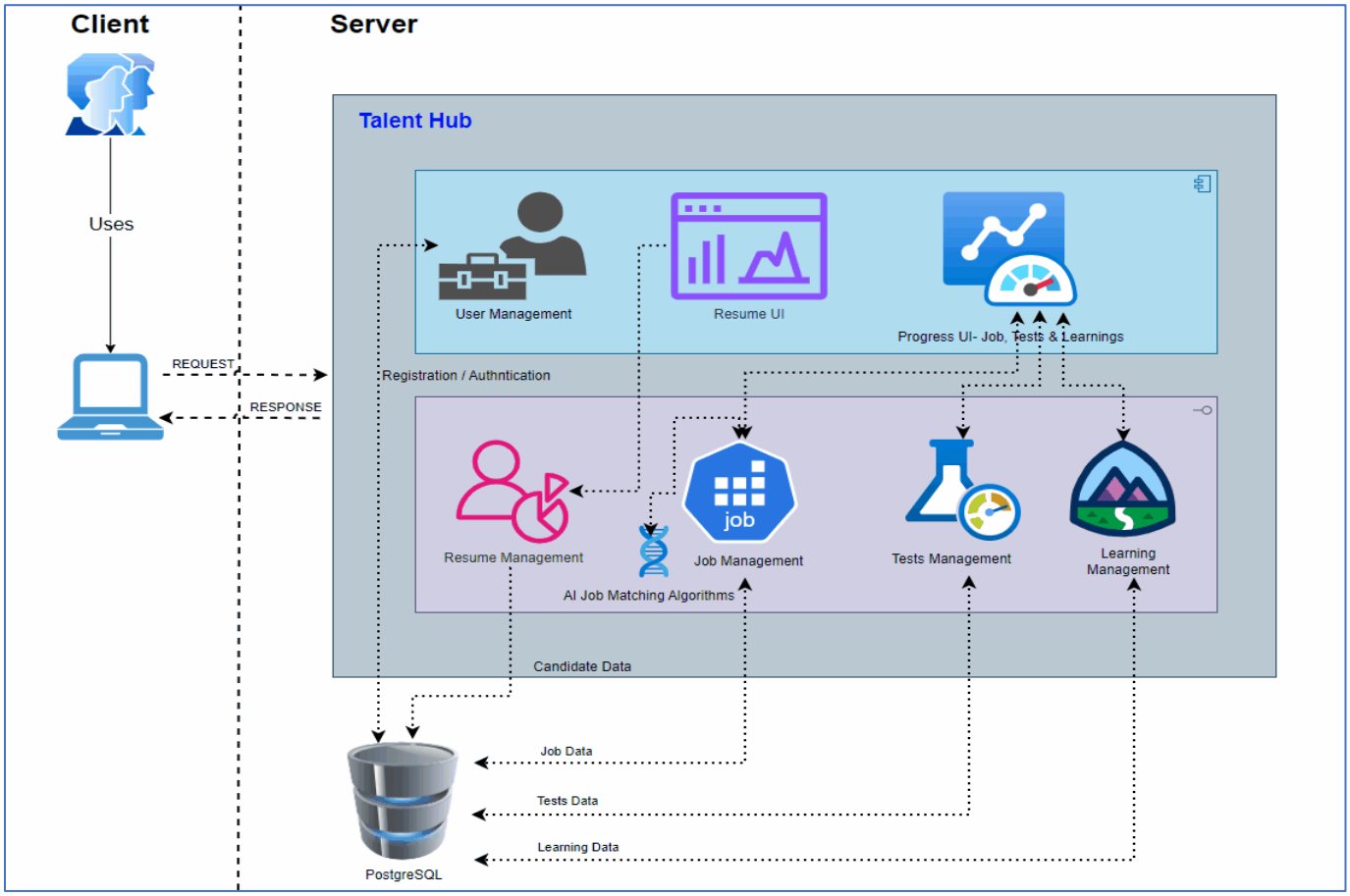
1. **Requirement Gathering and Analysis:** 1 month.
2. **Design and Architecture:** 1.5 months.

3. **Frontend and Backend Development:** 3 months.
4. **AI/ML Model Development:** 2 months.
5. **Testing and Quality Assurance:** 1.5 months.
6. **Deployment and Launch:** 1 month.

Possible Architecture and Design Patterns:

Component	Architecture	Design Pattern
User Authentication & Access Control	Client-Server Architecture	OAuth 2.0
Candidate Registration & Profile Management	MVC Architecture	Singleton
Resume Parsing	Microservices Architecture	Adapter
Job Matching Algorithm	Service-Oriented Architecture	Strategy
Testing Module	Layered Architecture	Observer
Learning Recommendation Engine	Event-Driven Architecture	Factory
Database Management	Database Sharding	Repository
Scalability & Performance	Cloud-Native Architecture	Caching with Redis
Frontend	Component-Based Architect.	Flux
Backend	Hexagonal Architecture	Dependency Injection
AI Module	Event-Driven Microservices	Chain of Responsibility
Security Module	Defense-in-Depth Security	Builder
Overall System	Microservices with Kubernetes	Circuit Breaker

Systemic View:



DB Tables:

Table Name	Columns	Relationships
Users	user_id (PK), email, password, user_type	NA
Candidate Profile	candidate_id (PK), user_id (FK), name, education, experience, skills	One-to-One with Users
Company Profile	company_id (PK), user_id (FK), name, industry, size, description	One-to-One with Users
Resumes	resume_id (PK), candidate_id (FK), resume_file_path, parsed_information	One-to-Many with Candidate Profile
Jobs	job_id (PK), company_id (FK), title, description, requirements, location, salary	One-to-Many with Company Profile
Tests	test_id (PK), job_id (FK), test_type, duration, difficulty_level	One-to-Many with Jobs
Test Results	result_id (PK), test_id (FK), candidate_id (FK), score, status	One-to-Many with Tests
Learning Programs	program_id (PK), candidate_id (FK), program_name, provider, duration, description	One-to-Many with Candidate Profile
UserSessions	session_id (PK), user_id (FK), expiry_time	NA
UserRole	role_id (PK), role_name	NA

High-Level User Stories (Epics):

User Story ID	User Story Title	User Story Description
1	Candidate Registration and Profile Creation	As a candidate, I want to register on the platform and create a profile so that I can apply for job openings.
2	Resume Uploading and Parsing	As a candidate, I want to upload my resume, and have it parsed so that my skills and experience can be added to my profile.
3	Job Search and Matching	As a candidate, I want to search for job openings and see relevant job matches based on my profile.
4	Testing Rounds	As a candidate, I want to participate in testing rounds to demonstrate my skills to potential employers.
5	Learning Recommendations	As a candidate, I want to receive personalized learning recommendations to improve my skills and increase my chances of getting hired.

Low-Level User Stories (Functionality):

1. Candidate Registration and Profile Creation:

User Story ID	User Story Description	Acceptance Criteria	Size
1.1(FR)	As a new candidate, I want to register on the platform with my email and password.	User can successfully register with a valid email and password.	Small
1.2 (FR)	As a registered candidate, I want to fill in my profile details (e.g., education, experience, skills).	User can successfully fill in their profile details.	Medium
1.3 (NFR)	As a candidate, I want my password to be securely stored (e.g., hashed) to protect my account.	Passwords are securely stored using a secure hashing algorithm.	Small
1.4 (NFR)	As a candidate, I want the registration process to be fast and responsive.	Registration process completes within <#> minute.	Small

2. Resume Uploading and Parsing:

User Story ID	User Story Description	Acceptance Criteria	Size
2.1	As a candidate, I want to upload my resume in PDF or document format.	User can successfully upload a resume in PDF or document format.	Small
2.2	As a candidate, I want to see the parsed information from my resume (e.g., skills, experience).	Parsed information is displayed accurately for user review.	Medium

2.3 (NFR)	As a candidate, I want the resume parsing process to be fast, even for large resumes.	Resume parsing completes within <#> seconds for a <#>-page resume.	Small
2.4 (NFR)	As a candidate, I want clear feedback on the parsing process.	User receives a confirmation message after successful parsing.	Small

3. Job Search and Matching:

User Story ID	User Story Description	Acceptance Criteria	Size
3.1	As a candidate, I want to search for job openings based on criteria such as location, industry, and job type.	User can search for job openings based on specified criteria.	Small
3.2	As a candidate, I want to see job openings that match my profile and skills.	Search results include job openings that match the user's profile and skills.	Medium
3.3 (NFR)	As a candidate, I want the search results to be displayed quickly.	Search results are displayed within <#> seconds.	Small
3.4 (NFR)	As a candidate, I want the job descriptions to be displayed accurately and clearly.	Job descriptions are displayed accurately and clearly.	Small

4. Testing Rounds:

User Story ID	User Story Description	Acceptance Criteria	Size
4.1	As a candidate, I want to access and complete testing rounds.	User can access and complete testing rounds.	Medium
4.2	As a candidate, I want to receive feedback on my test results.	Test results are evaluated, and feedback is provided to the user.	Small
4.3 (NFR)	As a candidate, I want the testing platform to be reliable and available during scheduled testing times.	Testing platform is available 99.9% of the time during scheduled testing times.	Small
4.4 (NFR)	As a candidate, I want the testing platform to be able to handle multiple concurrent users.	Testing platform can handle <#> concurrent users.	Medium

5. Learning Recommendations:

User Story ID	User Story Description	Acceptance Criteria	Size
5.1	As a candidate, I want to view personalized learning recommendations.	User can view personalized learning recommendations.	Small
5.2	As a candidate, I want to enroll in recommended programs.	User can enroll in recommended programs.	Small

5.3 (NFR)	As a candidate, I want the recommendations to be relevant and aligned with my skills and areas for improvement.	Recommendations are relevant and aligned with the user's skills and areas for improvement.	Small
5.4 (NFR)	As a candidate, I want the recommendations to be presented clearly and be easy to understand.	Recommendations are presented clearly and are easy to understand.	Small