**Title for Portal:** Talent Hub

**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.

|  |  |  |
| --- | --- | --- |
| **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 |

Overall, Talent Hub aims to revolutionize the candidate selection process by leveraging AI-driven insights and personalized recommendations while ensuring secure access, data protection, and optimal performance.

**1. Candidate Registration and Profile Creation:**

**Functional Requirements:**

* Users should be able to register with the portal by providing basic information such as name, email, and password.
* Upon registration, users should be able to create and update their profiles, including personal details, contact information, work experience, education, and skills.
* The system should validate user input and ensure that mandatory fields are filled.

**Acceptance Criteria:**

* Users can successfully register and log in to the portal.
* Users can create and update their profiles with accurate information.
* The system prevents users from registering with incomplete or invalid information.

**2. Resume Uploading and Parsing:**

**Functional Requirements:**

* Users should be able to upload their resumes in PDF or document format.
* The system should parse the uploaded resumes to extract relevant information such as skills, experience, and education.
* The parsed information should be stored in the candidate's profile for use in job matching.

**Acceptance Criteria:**

* Users can successfully upload resumes in supported formats.
* The system parses resume accurately and extracts relevant information.
* Parsed information is stored correctly in the candidate's profile.

**3. Job Search and Matching:**

**Functional Requirements:**

* Users should be able to search for job openings based on criteria such as job title, location, and skills.
* The system should match candidates with relevant job openings based on their profiles and skills.
* Users should be able to view details of matched job openings and apply for them.

**Acceptance Criteria:**

* Users can search for job openings and see relevant results.
* The system accurately matches candidates with job openings based on their profiles.
* Users can view job details and successfully apply for matching positions.

**4. Testing Rounds:**

**Functional Requirements:**

* The system should conduct initial testing rounds to assess candidates' skills.
* Testing rounds may include multiple-choice questions, coding exercises, or other forms of assessment.
* Candidates should receive feedback on their performance in the testing rounds.

**Acceptance Criteria:**

* The system conducts testing rounds as per the defined assessment criteria.
* Candidates can complete testing rounds and receive feedback.
* Testing results are stored for future reference.

**5. Learning Recommendations:**

**Functional Requirements:**

* Based on the results of testing rounds, the system should recommend learning programs or courses to candidates.
* Recommendations should be tailored to the candidate's skill gaps and areas of improvement.
* Candidates should be able to view and enroll in recommended learning programs.

**Acceptance Criteria:**

* The system generates accurate learning recommendations based on testing results.
* Candidates can view and enroll in recommended learning programs.
* Recommended programs align with the candidate's skill gaps and learning needs.

These detailed requirements specifications outline the functional requirements and acceptance criteria for each of the high-level requirements related to candidate registration, resume parsing, job search and matching, testing rounds, and learning recommendations.

**1. Candidate Registration and Profile Creation**

**Functional Requirements:**

1. Allow candidates to register by providing basic information (name, email, password).
2. Validate email format and ensure uniqueness.
3. Provide options for candidates to fill in additional profile details (e.g., education, experience, skills).
4. Enable candidates to edit their profiles after registration.
5. Display a confirmation message upon successful registration.

**Non-Functional Requirements:**

1. Security: Ensure secure storage of user passwords (e.g., hashing).
2. Performance: Registration process should be fast and responsive.
3. Usability: Provide clear instructions and guidance during the registration process.

**Acceptance Criteria:**

1. User can successfully register with a valid email and password.
2. User can log in after registration using the provided credentials.
3. User can view and edit their profile information after registration.

**2. Resume Uploading and Parsing**

**Functional Requirements:**

1. Allow candidates to upload resumes in PDF or document format.
2. Parse resumes to extract relevant information (e.g., skills, experience, education).
3. Display parsed information to candidates for review and editing.
4. Provide an option to save parsed information to the candidate's profile.

**Non-Functional Requirements:**

1. Accuracy: Resume parsing should accurately extract information from resumes.
2. Performance: Parsing should be fast, even for large resumes.
3. Usability: Provide clear feedback to candidates on the parsing process.

**Acceptance Criteria:**

1. User can successfully upload a resume in PDF or document format.
2. Parsed information is displayed accurately for user review.
3. User can save parsed information to their profile.

**3. Job Search and Matching**

**Functional Requirements:**

1. Provide a search interface for candidates to search for job openings.
2. Allow candidates to filter job search results based on criteria such as location, industry, and job type.
3. Display job openings that match the candidate's profile and skills.
4. Provide detailed job descriptions for each job opening.

**Non-Functional Requirements:**

1. Performance: Search results should be displayed quickly.
2. Accuracy: Job matching should accurately match candidates with relevant job openings.
3. Usability: Search interface should be intuitive and easy to use.

**Acceptance Criteria:**

1. User can search for job openings based on specified criteria.
2. Search results include job openings that match the user's profile and skills.
3. Job descriptions are displayed accurately and clearly.

**4. Testing Rounds**

**Functional Requirements:**

1. Conduct initial testing rounds for candidates to demonstrate their skills.
2. Provide different types of tests (e.g., multiple-choice, coding exercises) to assess different skills.
3. Evaluate test results and provide feedback to candidates.

**Non-Functional Requirements:**

1. Reliability: Testing platform should be reliable and available during scheduled testing times.
2. Security: Ensure test content is secure and not accessible to unauthorized users.
3. Performance: Testing platform should be able to handle multiple concurrent users.

**Acceptance Criteria:**

1. User can access and complete testing rounds.
2. Test results are evaluated, and feedback is provided to the user.

**5. Learning Recommendations**

**Functional Requirements:**

1. Recommend learning programs or courses based on the candidate's skills and areas for improvement.
2. Provide information about recommended programs (e.g., course content, duration, provider).
3. Allow candidates to enroll in recommended programs.

**Non-Functional Requirements:**

1. Personalization: Recommendations should be personalized based on the candidate's profile and test results.
2. Accuracy: Recommendations should align with the candidate's skills and areas for improvement.
3. Usability: Recommendations should be presented clearly and be easy to understand.

**Acceptance Criteria:**

1. User can view personalized learning recommendations.
2. Recommendations are relevant and aligned with the user's skills and areas for improvement.
3. User can enroll in recommended programs.

These detailed requirements specifications outline the functional and non-functional requirements for each of the high-level requirements, along with acceptance criteria to validate the implementation of these requirements.

**Epics or High-Level User Stories:**

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:**

Candidate Registration and Profile Creation

**Function:**

* **Story:** As a new candidate, I want to register on the platform with my email and password.
  + **Acceptance Criteria:** User can successfully register with a valid email and password.
  + **Size:** Small
* **Story:** As a registered candidate, I want to fill in my profile details (e.g., education, experience, skills).
  + **Acceptance Criteria:** User can successfully fill in their profile details.
  + **Size:** Medium

**Non-Functional:**

* **Story:** As a candidate, I want my password to be securely stored (e.g., hashed) to protect my account.
  + **Acceptance Criteria:** Passwords are securely stored using a secure hashing algorithm.
  + **Size:** Small
* **Story:** As a candidate, I want the registration process to be fast and responsive.
  + **Acceptance Criteria:** Registration process completes within 1 minute.
  + **Size:** Small

Resume Uploading and Parsing

**Function:**

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

**Non-Functional:**

* **Story:** As a candidate, I want the resume parsing process to be fast, even for large resumes.
  + **Acceptance Criteria:** Resume parsing completes within 5 seconds for a 2-page resume.
  + **Size:** Small
* **Story:** As a candidate, I want clear feedback on the parsing process.
  + **Acceptance Criteria:** User receives a confirmation message after successful parsing.
  + **Size:** Small

Job Search and Matching

**Function:**

* **Story:** As a candidate, I want to search for job openings based on criteria such as location, industry, and job type.
  + **Acceptance Criteria:** User can search for job openings based on specified criteria.
  + **Size:** Small
* **Story:** As a candidate, I want to see job openings that match my profile and skills.
  + **Acceptance Criteria:** Search results include job openings that match the user's profile and skills.
  + **Size:** Medium

**Non-Functional:**

* **Story:** As a candidate, I want the search results to be displayed quickly.
  + **Acceptance Criteria:** Search results are displayed within 2 seconds.
  + **Size:** Small
* **Story:** As a candidate, I want the job descriptions to be displayed accurately and clearly.
  + **Acceptance Criteria:** Job descriptions are displayed accurately and clearly.
  + **Size:** Small

Testing Rounds

**Function:**

* **Story:** As a candidate, I want to access and complete testing rounds.
  + **Acceptance Criteria:** User can access and complete testing rounds.
  + **Size:** Medium
* **Story:** As a candidate, I want to receive feedback on my test results.
  + **Acceptance Criteria:** Test results are evaluated, and feedback is provided to the user.
  + **Size:** Small

**Non-Functional:**

* **Story:** As a candidate, I want the testing platform to be reliable and available during scheduled testing times.
  + **Acceptance Criteria:** Testing platform is available 99.9% of the time during scheduled testing times.
  + **Size:** Small
* **Story:** As a candidate, I want the testing platform to be able to handle multiple concurrent users.
  + **Acceptance Criteria:** Testing platform can handle 1000 concurrent users.
  + **Size:** Medium

Learning Recommendations

**Function:**

* **Story:** As a candidate, I want to view personalized learning recommendations.
  + **Acceptance Criteria:** User can view personalized learning recommendations.
  + **Size:** Small
* **Story:** As a candidate, I want to enroll in recommended programs.
  + **Acceptance Criteria:** User can enroll in recommended programs.
  + **Size:** Small

**Non-Functional:**

* **Story:** As a candidate, I want the recommendations to be relevant and aligned with my skills and areas for improvement.
  + **Acceptance Criteria:** Recommendations are relevant and aligned with the user's skills and areas for improvement.
  + **Size:** Small
* **Story:** As a candidate, I want the recommendations to be presented clearly and be easy to understand.
  + **Acceptance Criteria:** Recommendations are presented clearly and are easy to understand.
  + **Size:** Small

These low-level user stories break down the functionality and non-functional aspects of each high-level requirement into smaller, manageable tasks with clear acceptance criteria and estimated sizes.

# **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 |

These low-level user stories provide a breakdown of the functionality, acceptance criteria, and estimated size for each component of the high-level requirements.