# 

**Project Overview**

* Project Name: Career Upskilling Agent
* Objective: AI platform helps programmers and IT students create personalized learning paths , aligned with career goals , technology trends and current
* Duration: From 26/05/2025 – To 13/06/2025
* Team Members:
* Nguyễn Quang Hiếu (captain)
* Hoàng Ngọc Linh
* Đoàn Thị Phương Thảo
* Phạm Hoàng Dương
* Bùi Văn Hiếu
* Nguyễn Đức Tùng
* Vũ Hồng Tuân
* Lưu Trường Sơn
* Nguyễn Đức Tùng

## 

## Preface

We would like to extend our sincerest gratitude to everyone who provided us with the opportunity to undertake this "Career Upskilling Agent" project. In particular, we express our deep appreciation to the Organizing Committee (OC) for their continuous support, guidance, and companionship throughout the development process. This invaluable support has been a tremendous motivation, helping us to complete this System Requirements Specification (SRS) document to the best of our abilities.

## Summary

This System Requirements Specification (SRS) document details the functional and non-functional requirements for the "Career Upskilling Agent" system. The primary objective of this project is to build an AI-powered personal learning agent capable of creating highly customized learning paths using user inputs and dynamic public resources.

The system will provide a platform that allows users to input their current skills, target proficiency levels, and time constraints. Based on this information, the system will generate a structured, personalized learning roadmap (in Gantt chart or Mindmap format), suggest personalized learning resource modules (e.g., Coursera, Udemy), and enable users to track their learning progress through an intuitive dashboard. The learning roadmap will automatically adjust based on user progress and learning style. The system will also support downloading the roadmap in various formats (PDF/Excel).

Technologically, the system will employ a Microservices architecture with a user interface developed using React and Tailwind CSS, and backend services utilizing Spring Boot (Java) and Python. AI models such as Gemini and Hugging Face will be integrated for input analysis, roadmap generation, and RAG (Retrieval-Augmented Generation) implementation. A PostgreSQL database with pgvector will be used for data storage. The system will be deployed on AWS, ensuring scalability, security, and high availability.

Currently, the project is in the "Project Management & Progress" phase, having completed the master plan, task board, and adopted the Scrum Framework methodology.

# Index

**Image catalog**

[Image 1](#_fhvq8ti7df2q) 11

[Image 2](#_eeimwfrujfxq) 12

# **System Requirement Specification (SRS) Document - Career Upskilling Agent**

## 1. Introduction

### 1.1 Purpose

This document outlines the functional and non-functional requirements for the "Career Upskilling Agent" system. The primary goal is to build an AI-powered personal learning agent that creates highly customized learning paths using user inputs and dynamic public resources.

### 1.2 Product Scope

The system will provide a platform that allows users to:

* Input current skills, target proficiency levels, and time constraints.
* Receive a structured learning roadmap (Gantt/Mindmap).
* Access personalized resource modules (e.g., Coursera, Udemy).
* Download the roadmap in various formats (PDF/Excel).
* Track learning progress through a dashboard.
* Automatically adjust the roadmap based on user progress and learning style.

### 1.3 Intended Audience and Conventions

This document is intended for developers, testers, project managers, and other stakeholders involved in the development and deployment of the "Career Upskilling Agent" system.

**Conventions:**

* **User:** An individual utilizing the system to create and manage learning roadmaps.
* **System:** The "Career Upskilling Agent" software.
* **AI:** The artificial intelligence models used within the system.

## 2. Overall Description

### 2.1 Product Perspective and Vision

In the context of rapid technological evolution requiring continuous upskilling, this system aims to address pain points such as scattered resources, difficulty in creating skill roadmaps, and time-consuming searches for quality content. The vision is to create an AI agent capable of dynamically generating personalized and adaptive learning roadmaps tailored to user needs.

### 2.2 User Functions

* **Goal Input:** Users can input their learning goals (e.g., "Learn Backend Development").
* **Time Setting:** Users can set deadlines and weekly time commitments.
* **Skill Assessment:** Users can leverage their profile or past roadmaps for skill assessment, or manually select relevant skills.
* **Skill Confirmation:** Users review and confirm their currently possessed skills.
* **Roadmap Viewing:** View personalized learning roadmaps as Gantt charts or Mindmaps.
* **Resource Access:** Access recommended resource modules.
* **Download:** Download roadmaps in PDF/Excel formats.
* **Progress Tracking:** Monitor learning progress via a dashboard.

### 2.3 AI Capabilities and Characteristics

* **User Input Analysis:** AI analyzes user's current skills, target levels, and time constraints.
* **Roadmap Design:** AI designs a structured learning roadmap.
* **Content Recommendation:** AI recommends personalized learning content.
* **Adaptive Adjustment:** AI adjusts the roadmap based on user learning style and progress.
* **Progress Monitoring:** AI assists in tracking and adapting progress.
* **Report Generation:** AI is capable of generating reports.
* **RAG (Retrieval-Augmented Generation):**
  + **Retrieval:** AI searches for the latest and most relevant external resources.
  + **Generation:** AI combines retrieved data to create updated, context-aware learning plans.

### 2.4 Operating Environment

* **System Architecture:**
  + **Front-end:** React, Tailwind CSS.
  + **Back-end:** Spring Boots Java, Python.
  + **AI Model:** Gemini, Hugging Face MiniLM. Llama 3 70B.
  + **Database:** PostgreSQL RDBMS, PostgreSQL PGVector.
  + **Deployment:** EC2, S3, RDS, Load Balancer, Vercel.
  + **CI/CD:** Git (Version Control), Build/CI/CD runner (Jenkins/Travis CI/GitLab CI), Docker, Vercel.
* **AI-related Tools:**
  + Gamma AI: Slide/Presentation Generation, Documentation Sketches.
  + GitHub Copilot: Code Suggestion, Refactoring, Automation.
  + Helicone: Cost Monitoring, Token Usage Tracking, Prompt Debugging.
  + Diagram GPT: AI Diagram Generation, System/Architecture Mapping.
  + ChatGPT: Brainstorming, Prompt Writing, Documentation, Code Help.
  + OpenAI Tokenizer: Token Estimation, Prompt Length Control.

## 3. Functional Requirements

### 3.1 FR1: User Input Management

* **FR1.1 Learning Goal Input:** The system shall allow users to input their primary learning goal (e.g., "Become a Full-stack Developer").
* **FR1.2 Time Constraint Setting:** The system shall allow users to set deadlines (e.g., "3 months") and weekly time commitments (e.g., "10 hours/week").
* **FR1.3 Current Skill Assessment:**
  + **FR1.3.1 Use Profile/Past Roadmap:** The system shall allow users to leverage their profile or previously generated roadmaps for skill assessment.
  + **FR1.3.2 Manual Skill Selection:** The system shall allow users to manually select relevant skills they currently possess.
* **FR1.4 Skill Confirmation:** The system shall display a summary of assessed skills and allow users to confirm or modify them.

### 3.2 FR2: Learning Roadmap Generation and Management

* **FR2.1 Personalized Roadmap Generation:** The system shall generate a structured, personalized learning roadmap based on user's goals, time constraints, and existing skills.
* **FR2.2 Roadmap Visualization:** The system shall display the learning roadmap in a visual format (Gantt chart or Mindmap).
* **FR2.3 Resource Recommendation:** The system shall recommend relevant resource modules (e.g., Coursera, Udemy courses) for each segment of the roadmap.
* **FR2.4 Dynamic Roadmap Update:** The system shall automatically update and adjust the learning roadmap based on user's actual progress and learning style.

### 3.3 FR3: Progress Tracking

* **FR3.1 Progress Dashboard:** The system shall provide a visual dashboard for users to track their learning progress.
* **FR3.2 Completion Marking:** The system shall allow users to mark modules or skills as completed.

### 3.4 FR4: Export and Sharing

* **FR4.1 Roadmap Download:** The system shall allow users to download their learning roadmap in common formats such as PDF and Excel.

## 4. Non-Functional Requirements

### 4.1 Performance

* **NFR4.1.1 Response Time:** The system shall generate a learning roadmap within X seconds (e.g., 10 seconds) after receiving user input.
* **NFR4.1.2 Scalability:** The system shall be capable of supporting Y concurrent users (e.g., 10,000 users) without significant performance degradation.
* **NFR4.1.3 Resource Retrieval Time:** The time taken to retrieve external resources via RAG shall be efficient, not causing significant delays in roadmap generation.

### 4.2 Security

* **NFR4.2.1 User Authentication:** The system shall implement secure authentication mechanisms to protect user data.
* **NFR4.2.2 Data Authorization:** Access to user data shall be strictly controlled based on user roles.
* **NFR4.2.3 Data Protection:** User data, including personal information and learning roadmaps, shall be encrypted both in transit and at rest.

### 4.3 Usability

* **NFR4.3.1 Intuitive User Interface:** The user interface shall be intuitive, easy to use, and navigate.
* **NFR4.3.2 Responsive Design:** The system shall function optimally across various devices (desktops, tablets, mobile phones) and screen sizes.
* **NFR4.3.3 Feedback and Notifications:** The system shall provide clear and timely feedback for user actions and system events.

### 4.4 Reliability

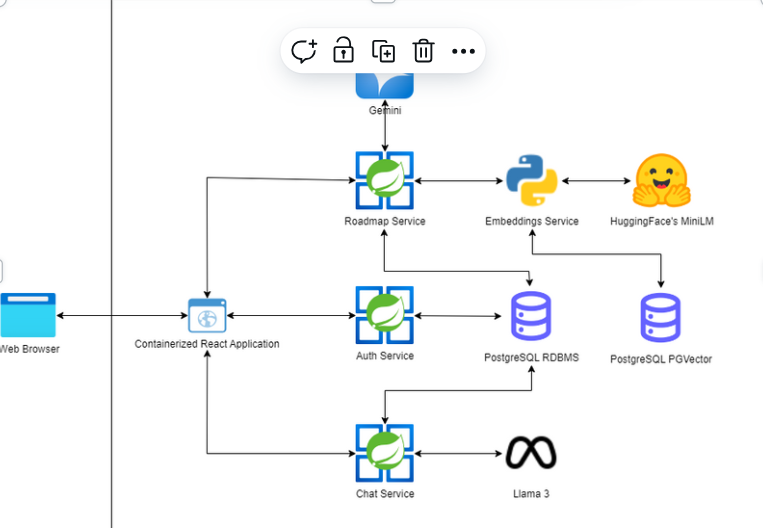
* **NFR4.4.1 Availability:** The system shall be available 99.9% of the time.
* **NFR4.4.2 Fault Recovery:** The system shall have mechanisms to gracefully recover from failures and minimize data loss.
* **NFR4.4.3 Data Consistency:** User roadmap and progress data shall remain consistent across all sessions and devices.

### 4.5 Maintainability

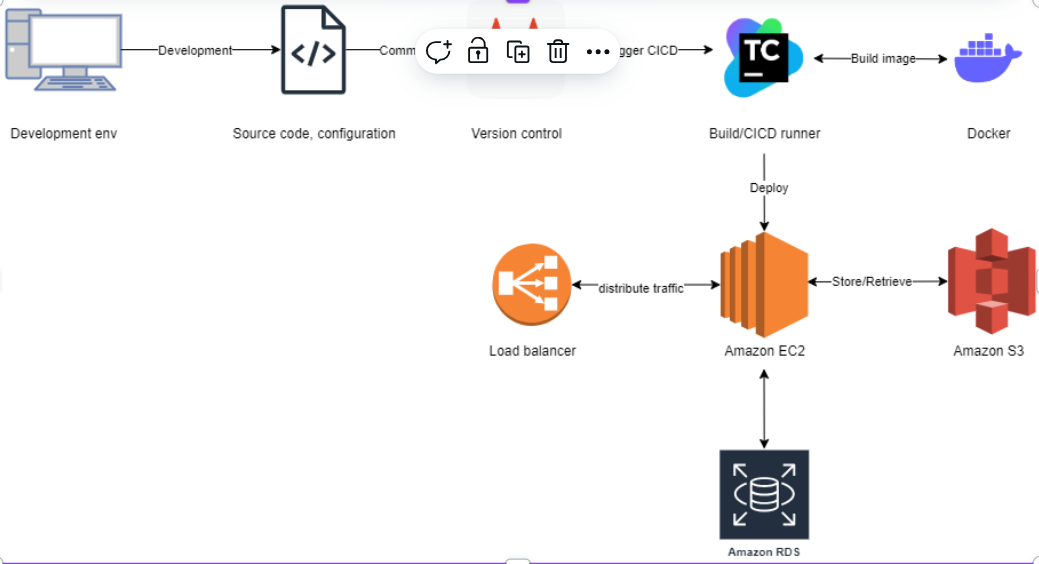
* **NFR4.5.1 Modularity:** The system shall be designed with modularity to facilitate updates and maintenance.
* **NFR4.5.2 Readable Code:** The source code shall be well-commented and adhere to coding standards.
* **NFR4.5.3 Logging and Monitoring:** The system shall have robust logging and monitoring mechanisms for debugging and performance analysis.

## 5. System Architecture and Technology

### 5.1 System Architecture



### Image 1



### Image 2

A more detailed description of the architecture diagram, if available, with data flows.

The system will follow a Microservices architecture to ensure scalability and maintainability. Key components include:

* **User Interface (UI):** A web-based user interface developed using React and Tailwind CSS.
* **Backend Services:** Services developed using Spring Boot (Java) and Python, handling business logic, AI integration, and database interactions.
* **AI Core:** AI models leveraging Gemini and Hugging Face for input analysis, roadmap generation, and RAG implementation.
* **Database:** PostgreSQL used as the primary database, with pgvector for handling vector embeddings for AI-related operations.
* **Deployment Infrastructure:** Deployed on AWS using Amazon EC2 for application servers, Amazon S3 for static storage, and Amazon RDS for the PostgreSQL database.
* **Load Balancing:** Ensures optimal traffic distribution and high availability.

### 5.2 Technology Stack

* **Front-end:** React, Tailwind CSS
* **Back-end:** Java (Spring Boot), Python
* **AI Model:** Gemini, Hugging Face
* **Database:** PostgreSQL, pgvector
* **Deployment:** AWS (EC2, S3, RDS), Docker
* **Version Control:** Git
* **CI/CD:** Jenkins/Travis CI/GitLab CI (though the slide only generically states "Build/CI/CD runner")

## 6. Development Roadmap and Progress

### 6.1 Development Phases

* **Stage 1 - Requirement & System Design:**
  + Analyze SRS and define MVP.
  + Identify key inputs/outputs.
  + Design system architecture and database.
  + Create dashboard and project plan.
* **Stage 2 - Prototype Development:**
  + Create input form and initial UI.
  + Generate static learning roadmaps (without AI).
  + Implement simple roadmap visualization.
* **Stage 3 - Enhancement & Integration:**
  + Integrate RAG (Retrieval-Augmented Generation)-based AI model.
  + Add progress tracking and reporting features.
  + Improve roadmap visualization.
  + Finalize UI.

### 6.2 Current Progress

The project has reached the "Project Management & Progress" phase, including a master plan, task board, and Scrum Framework methodology.

**References**

1. OpenAI. GPT-4 Technical Report. [Online]. Available: <https://openai.com>. Accessed: 2024.
2. Anthropic. Claude 3 Model Overview. [Online]. Available: <https://www.anthropic.com>. Accessed: 2024.
3. Google DeepMind. Gemini 1.5 Release and Capabilities. [Online]. Available: <https://ai.google>. Accessed: 2024.
4. Mistral AI. Mistral & Mixtral Models Documentation. [Online]. Available: <https://mistral.ai>. Accessed: 2024.
5. Meta AI. LLaMA 3: Open Foundation Models by Meta. [Online]. Available: <https://ai.meta.com>. Accessed: 2024.
6. Cohere. Embedding and Reranking Models. [Online]. Available: <https://cohere.ai>. Accessed: 2023.
7. Amazon Web Services. AWS Bedrock and Redshift Integration Guide. [Online]. Available:<https://aws.amazon.com>. Accessed: 2024.
8. Grafana Labs. Grafana Documentation and Tutorials. [Online]. Available: <https://grafana.com>. Accessed: 2024.
9. Helicone AI. Token Usage Tracking for LLMs. [Online]. Available: <https://www.helicone.ai>. Accessed: 2024.
10. Notion. Notion AI: Your Creative Workspace with AI Assistance. [Online]. Available: [https://www.notion.so/product/ai.](https://www.notion.so/product/ai) Accessed: 2024.
11. Canva. Magic Studio for AI-Powered Presentations. [Online]. Available: <https://www.canva.com/magic>. Accessed: 2024.
12. Eraser.io. DiagramGPT: AI Flowchart and Diagram Generator. [Online]. Available: <https://eraser.io>. Accessed: 2024.