

# Final Year Project – Requirements Analysis Canvas

---

## 1. Problem Understanding (WHY)

### Problem Statement:

Describe the core problem your target users are facing. Focus on: - What users are trying to achieve - Why existing solutions (courses, generic AI tools, platforms) are insufficient - The gap between user expectations and current tools

(Write 1–2 clear paragraphs. No technical jargon.)

---

## 2. Target Users (WHO)

Define who will use the application.

### User Persona 1

- Skill level:
- Goal:
- Daily time availability:
- Pain points:

### User Persona 2

- Skill level:
- Goal:
- Daily time availability:
- Pain points:

(Optional: Add Persona 3 if needed)

---

## 3. Core Use-Cases (WHAT USERS WILL DO)

Describe the user journey step by step: 1. User opens the application 2. User creates an account / logs in 3. User defines the skill they want to learn 4. User receives a personalized roadmap 5. User completes tasks and receives feedback 6. User tracks progress over time

(Focus on user actions, not system internals.)

---

## 4. Functional Requirements (SYSTEM FEATURES)

List all essential features the system must provide: - User authentication (signup/login) - Skill input and validation - Roadmap generation and display - Task assignment and completion tracking - Feedback generation - Progress storage and retrieval

(Only include features required for a working system.)

---

## 5. Non-Functional Requirements (QUALITY ATTRIBUTES)

Define how the system should behave: - Performance: fast response for user actions - Reliability: system should not crash if AI fails - Usability: simple and beginner-friendly interface - Scalability: should support more users in future - Security: basic protection of user data

---

## 6. AI Role Definition (BOUNDARIES)

Clearly define the role of AI in the system:

**AI Will:** - Generate explanations and learning content - Adapt difficulty based on user progress - Provide feedback and reframing

**AI Will NOT:** - Control system logic - Decide progression rules - Replace core application flow

**AI Input:** - User skill description - User profile data - Progress data

**AI Output:** - Structured text (lessons, feedback, tasks)

---

## 7. Data Requirements (WHAT DATA IS STORED)

Identify the types of data the system needs to store: - User profile information - Selected skill and preferences - Generated roadmap structure - Task completion status - Feedback history

(No database design required at this stage.)

---

## 8. Constraints & Assumptions (REALITY CHECK)

List real-world limitations: - Limited development time - Beginner-level development team - Limited budget for AI API usage - Academic scope and deadlines - Internet dependency

---

## 9. Success Criteria (HOW SUCCESS IS MEASURED)

Define how you will know the project is successful: - Application can be demonstrated live - Users can generate a learning roadmap - Tasks and progress tracking work correctly - AI personalization is visible in demo - Supervisor approves scope and implementation

---

**Document Status:** Draft **Next Step:** Review with Supervisor → Finalize → Move to System Design