**Project Design Phase**

**Proposed Solution Template**

| Date | 30 July 2025 |
| --- | --- |
| Team ID | PNT2025TMID09555 |
| Project Name | The Roadmap Generator |
| Maximum Marks | 2 Marks |

## **Proposed Solution Document**

### **Project Title:**

The Roadmap Generator – Llama2 Powered Tool

### **Category:**

Generative AI / Personalized Learning

### **Objective:**

To develop a web-based application that generates a personalized, step-by-step learning roadmap for any user-defined skill using the Llama2 language model. The roadmap includes curated free resources and guidance tailored to individual goals.

### **Problem Statement:**

Learners often struggle to find structured guidance for acquiring new skills, resulting in wasted time and disorganized learning. There is a lack of tools that generate customized, progressive learning paths based on AI.

### **Proposed Solution:**

Develop a Streamlit-based web application where:

* Users input a desired skill (e.g., “Learn NLP”)
* The input is passed to a Llama2 model (via CTransformers)
* A personalized learning roadmap is generated using prompt engineering
* Free resources are recommended in a structured format (e.g., by phase or week)
* The result is displayed cleanly in a web interface

### **System Architecture:**

[User Input] --> [Streamlit UI] --> [Backend: Python + CTransformers]

--> [Llama2 Inference] --> [Roadmap Output] --> [Streamlit Display]

### **Tools & Technologies:**

| **Component** | **Technology** |
| --- | --- |
| Language Model | Llama2 (GGUF format) |
| Model Loader | CTransformers |
| Web App Framework | Streamlit |
| Environment Mgmt. | Python 3.10 + pip |
| Deployment | Streamlit Cloud / GitHub |
| Prompting Method | Custom zero-shot prompt |

### **Required Skills:**

* Regression & Classification Algorithms
* Natural Language Processing (NLP)
* Streamlit Web App Development
* Generative AI Prompt Engineering

### **Functional Flow:**

1. User Input: User types in a skill (e.g., “Deep Learning”)
2. Backend Processing:
   * Streamlit fetches input
   * Sends to Llama2 with custom prompt
   * Receives AI-generated roadmap
3. Display Output:
   * Roadmap is formatted and shown in app
   * Resource links and phases/week breakdown included

### **Folder Structure:**

roadmapper/

├── app.py

├── setup.sh

├── requirements.txt

├── README.md

├── .gitignore

├── utils/

│ └── model\_loader.py

└── model/

└── llama-2-7b-chat.Q3\_K\_M.gguf (auto-downloaded)

### **🚀 Deployment Plan:**

* Use setup.sh to auto-download the model
* Deploy to Streamlit Cloud via GitHub
* Maintain minimal GitHub repo (no large model files)

### **🔒 Limitations & Considerations:**

* Hugging Face model requires authentication (can be handled in CLI or token)
* Llama2 inference might be slow on CPU (consider quantized model)
* Output quality depends on prompt tuning

### **✅ Future Enhancements:**

* Add authentication & saved roadmaps
* Include difficulty level or timeline as inputs
* Use local embeddings for resource ranking