**Project Title**: Cyberskool Companion  
**Module**: AI Text Summarizer  
**Type**: Web API with AI model backend  
**Tech Stack**: Python (Flask, Hugging Face Transformers, PyTorch), Flask REST API, HTML/JSON interfaces

**1. Functional Requirements (Standard Features)**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Automatic Summarization | Generates concise summaries from input text |
| Extractive Summarization | Selects important sentences without altering them |
| Length Control | Option to choose short, medium, or long summary |
| Single Document Summarization | Works with one document at a time |
| Multi-language Support (basic) | Supports English (others optional via tokenizer support) |
| Document Format Support | Accepts .txt, .pdf, .docx, and raw string |
| API Endpoint for Summarization | REST endpoint to post input text and return summarized output |
| Export Summary | Returns response in downloadable TXT or JSON |
| Drag-and-Drop Upload (Optional Web UI) | Upload a file via frontend (optional phase) |
| Basic Security | Input sanitation, CORS, request size limits |

**🧱 2. Architectural Design**

**2.1 High-Level Architecture**

UI

Flask REST API

Pre-trained Transformers

Summarizer Model Engine

Text Preprocessing

**2.2 Modules Breakdown**

* **UI Layer**: Upload text/files, specify summary length
* **API Layer**: Flask app with routes for upload and summarization
* **Service Layer**: Summarization logic using Hugging Face transformers
* **Model Layer**: Pre-trained model (facebook/bart-large-cnn) loaded via transformers, executed via PyTorch