Essay and Assignment feedback :

Customized Resumes for Every Opportunity

Hackathon Project Phases Template that ensures students can complete it efficiently while

covering all six phases. The template is structured to capture essential information without

being time-consuming.

Hackathon Project Phases Template

Project Title: WriteWise:Essay and Assignment feedback

Team Name: Band of Hawks

Team Members:

- Vajra Nandhan
- Pavan Kumar.Jonnada
- Nithin Ganji
- Manogna Ram Peddi
- Aadarsh Goud

Phase-1: Brainstorming & Ideation

Objective:

The objective of this project is to improve essay and assignment feedback through structured brainstorming and ideation. It aims to enhance clarity, coherence, and argument strength by using techniques like mind mapping, peer review, and rubric-based evaluation. This approach fosters critical thinking and iterative refinement, leading to more effective learning outcomes

Key Points:

1. Problem Statement:

Many students struggle to understand and apply feedback for improvement, leading to stagnation in writing skills. This project aims to address these issues by incorporating structured brainstorming and ideation techniques to enhance clarity, coherence, and actionable insights in feedback.

2. Proposed Solution:

Develop an AI-powered platform that analyzes essays and assignments, providing instant feedback on grammar, coherence, structure, and originality. Implement NLP techniques to assess writing quality and suggest improvements. Include a plagiarism checker and a scoring system based on predefined rubrics. Offer real-time suggestions to enhance academic writing skills.

3. Target Users:

- **Students** – To receive instant feedback and improve their academic writing.
- **Teachers & Professors** – To streamline grading and provide detailed, AI-assisted feedback.
- **Educational Institutions** – To enhance learning outcomes and maintain academic integrity.
- **Content Writers & Researchers** – To refine writing quality and ensure originality.

4. Expected Outcome:

The project will deliver an AI-powered platform that provides instant, detailed feedback on essays and assignments, enhancing writing quality and academic performance. It will assist students in improving their work while reducing educators' grading workload through automated analysis and plagiarism detection.

Phase-2: Requirement Analysis

Objective: To create an intelligent feedback system that helps students and educators improve academic writing by offering real-time analysis, error detection, and personalized suggestions.

● Key Points:

- Analyze grammar, coherence, structure, and readability.
- Ensure originality and prevent academic misconduct.
- Provide real-time suggestions based on predefined rubrics.

Technical Requirements:

- **Programming Language:** Python (>=3.8)
- **Web Framework:** Streamlit (>=1.28.0)
- **NLP Libraries:**
- **Transformers (>=4.30.2)** – For AI models like DistilBERT and T5
- **Torch (>=2.0.1)** – For deep learning computations
- **NLTK (>=3.8.1)** – For tokenization and basic NLP processing
- **SentencePiece (>=0.1.99)** – For tokenization of input texts
- **NumPy (>=1.24.3)** – For numerical processing

Phase-3: Project Design

Objective: Define system architecture, user flow, and UI/UX components.

Key Points:

1. System Architecture Diagram: (Simple sketch or flowchart)

2. User Flow: (How a user will interact with the project)

Phase-4: Project Planning (Agile Methodologies)

Objective:

● Break down the tasks using Agile methodologies.

Key Points:

1. Sprint Planning: (Divide work into tasks for each team member)

2. Task Allocation:

Code:Nandhan.V,Manogna Ram.P

Document:Aadarsh, Pavan

Demo video and ppt:Nithin.g

Phase-5: Project Development.

Objective:

● Code the project and integrate components.

Key Points:

1. Technology Stack Used: PYTHON,BERT T5

- Development Process: Project Setup & Environment Configuration
- Installed dependencies (Streamlit, Transformers, Torch, NLTK).

- Configured SSL certificates to fix NLTK download issues.
- Model Selection & Preloading
- Used DistilBERT for grammar checking and T5 for text improvement.
- Cached models using st.cache_resource() for efficient performance.
- Frontend Development (Streamlit UI)
- Designed essay input area, analysis buttons, and result sections.
- Implemented sidebar settings to enable/disable different analysis options.
- Backend Development (NLP Processing & AI Inference)
- Integrated tokenization, grammar correction, and text suggestions using pre-trained models.
- Implemented coherence analysis by computing paragraph similarity.
- Testing & Optimization
- Conducted functional & performance testing on various input sizes.
- Optimized batch processing and model inference to improve response time.

2. Challenges & Fixes:

1. Model Loading Issues

- Issue: AI models failed to load due to incorrect paths or missing files.
- Fix: Verified paths and used os.path.exists() to check model availability before loading.

2. Slow Performance on Large Texts

- Issue: Essays with 2000+ words caused significant processing delays.
- Fix: Implemented batch tokenization and used GPU acceleration when available.

3. NLTK Resource Errors

- Issue: Tokenization failed due to missing Punkt & Stopwords corpora.
- Fix: Added a function to auto-download missing NLTK resources at startup.

4. UI Freezing on Multiple Requests

- Issue: Clicking "Analyze" multiple times caused lag.
- Fix: Used Streamlit session state to disable buttons while analysis was in progress.

5. Incorrect Grammar Suggestions

- Issue: T5-generated corrections were not always accurate.
- Fix: Fine-tuned text suggestions and added a confidence threshold to reject bad suggestions

Phase-6: Functional & Performance Testing

Objective:

The Functional Testing objective is to ensure accurate grammar checking, style analysis, and coherence evaluation

while providing a seamless user experience. The Performance Testing objective is to verify that the system processes long essays efficiently, with response times under 10 seconds for 2000+ words

Key Points:

1. Test Cases Executed:

- Accurate AI-Powered Analysis – Uses DistilBERT for grammar correction, T5 for text improvement, and NLP techniques for style & coherence evaluation.
- Optimized Performance & Scalability – Preloads models in Streamlit session state, supports batch processing, and handles large essays efficiently (2000+ words in <10s).
- Robust Error Handling & Security – Prevents crashes with input validation, secures text processing against special characters & injection attacks, and ensures smooth user experience.

2. Bug Fixes & Improvements:

1.Model Loading Issues

- Bug: AI models (DistilBERT, T5) fail to load on Streamlit startup.
- Fix: Verify model paths, use torch_dtype=torch.float16 to reduce memory usage, and pre-load models efficiently.
- Tokenization Errors

- Bug: Errors like IndexError: list index out of range or RuntimeError: The size of tensor occur during tokenization.
- Fix: Use truncation=True, padding=True when tokenizing text to handle varying input sizes.
- 3Streamlit Session State Issues
- Bug: Buttons remain disabled even after models are loaded.
- Fix: Ensure models are set in session state inside the thread function after loading.

3. Final Validation: YES, it has reached our expectations.

4. Deployment (if applicable): (Hosting details or final demo link)

Final Submission

1. Project Report Based on the templates
2. Demo Video (3-5 Minutes)
3. Github link : https://github.com/nandhanvajra/WriteWise
4. Presentation