Hackathon Project Phases Template for the WriteWiseApp project.

# Hackathon Project Phases Template

**Project Title:**

WriteWise: Essay and Assignment Feedback Toolusing Bert and T5

**Team Name:**

(APEX)

**Team Members:**

* NivedithaThodishetty
* B Pavan Naidu
* G Nikhil Naidu
* P Nikitha

th

## Phase-1: Brainstorming & Ideation

**Objective:**

Students can use the tool to identify and correct grammatical errors and improve their writing style, resulting in more polished and professional essays.

**Key Points:**

 **Advanced NLP Integration**: The tool utilizes advanced NLP models like BERT for grammatical analysis and T5 for providing tailored suggestions for improving the writing.

 **Immediate Feedback**: It delivers quick, real-time feedback on essays and assignments, helping users to improve their writing right away.

 **Readability and Vocabulary Analysis**: The tool evaluates the readability of the text and assesses vocabulary complexity to ensure the writing is clear and appropriately advanced.

 **Comprehensive Evaluation**: WriteWise offers a holistic review by combining grammar correction, stylistic improvements, and content assessment, giving users a well-rounded evaluation of their work.

## Phase-2: Requirement Analysis

**Objective:**

To identify and correct grammatical errors and improve their writing style, resulting in more polished and professional essays.

**Key Points:**

 **Understanding Stakeholder Needs**: Requirement analysis involves gathering and understanding the needs, expectations, and objectives of stakeholders (clients, users, etc.) to ensure the project aligns with their goals.

 **Defining Functional and Non-Functional Requirements**: It clearly distinguishes between functional requirements (what the system should do) and non-functional requirements (how the system should perform, such as security or scalability).

 **Creating Detailed Documentation**: The process produces thorough documentation, including requirements specifications, which serve as a reference point throughout the project’s lifecycle to avoid scope creep and ensure all requirements are met.

 **Ensuring Feasibility**: During requirement analysis, the feasibility of the project is assessed to ensure that the requirements are realistic and achievable within the given constraints, such as time, budget, and technology.

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.

**Key Points:**

** Blueprint Creation: Project design involves creating a detailed blueprint or plan that outlines how the system or product will function and be structured, addressing both the technical and functional aspects.**

** Architecture and Structure: It defines the overall architecture of the project, including system components, their relationships, and how they will interact, ensuring scalability, maintainability, and performance.**

** Resource Planning: Project design also includes identifying the resources required, such as tools, technologies, personnel, and time, to ensure the project can be successfully developed and delivered.**

** Risk Mitigation: It anticipates potential risks and challenges, designing solutions or strategies to minimize these risks and ensuring the project remains on track for successful completion.**

## Phase-4: Project Planning (WriteWise AI)

**Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected**  **Outcome** |
| Sprint 1 | Environment Setup  & API Integration | 🔴 High | 6 hours  (Day 1) | End of Day  1 | Niveditha | Google API Key,  Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | Nikitha | API response format finalized | Basic UI with input fields |
| Sprint 2 | Vehicle Search &  Comparison | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Pavan | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Nikhil | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Pavan & Nikhil | API response, UI layout completed | Responsive UI, better user experience |
| Sprint 3 | Final Presentation  & Deployment | 🟢 Low | 1 hour  (Day 2) | End of Day  2 | Entire Team | Working prototype | Demo-ready project |

### ****Sprint 1 – Setup & Integration (Day 1)****

* **(🔴 High Priority)** Set up the development environment and install required libraries (Python, Streamlit, etc.).
* **(🔴 High Priority)** Integrate the WriteWise AI API for grammar analysis and suggestions.
* **(🟡 Medium Priority)** Design and build a basic UI with input fields for essay submission.

### ****Sprint 2 – Core Features & Debugging (Day 2)****

* **(🔴 High Priority)** Implement essay analysis and improvement features using WriteWise AI’s suggestions (grammar, readability, vocabulary).
* **(🔴 High Priority)** Debug and handle any API-related errors and ensure smooth integration between the front end and backend.

### ****Sprint 3 – Testing, Enhancements & Submission (Day 2)****

* **(🟡 Medium Priority)** Test API responses, refine UI for better usability, and fix UI bugs (e.g., responsiveness, input validation).
* **(🟢 Low Priority)** Prepare and finalize the demo for deployment, ensuring the system is ready for presentation.

## Phase-5: Project Development

**Objective:**

Implement core features of the AutoSage App.

### ****1. Technology Stack Used:****

* **Frontend**: Streamlit
* **Backend**: Google Gemini Flash API
* **Programming Language**: Python

### ****2. Development Process:****

* **Implement API Key Authentication and Gemini API Integration**: Securely integrate the Google Gemini Flash API to ensure reliable and authenticated feedback generation for grammar, readability, and vocabulary.
* **Develop Essay Feedback Logic**: Build core functionality that provides detailed analysis, grammar corrections, and style suggestions using the WriteWise AI API.
* **Optimize Feedback Generation for Performance and Relevance**: Enhance the accuracy and speed of feedback, ensuring the tool delivers contextually relevant suggestions to improve the user’s writing.

### ****3. Challenges & Fixes:****

* **Challenge**: Delayed API Response Times  
  **Fix**: Implement caching strategies to store frequently analyzed essays or common writing patterns, improving response time and reducing server load.
* **Challenge**: Limited API Calls Per Minute  
  **Fix**: Optimize API queries to only request essential data, avoiding unnecessary calls and ensuring efficient use of available API limits.

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the WriteWise AI works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Query: "Improve the grammar of this essay about climate change." | Relevant budget cars should be displayed. | ✅ Passed | Tester 1 |
| TC-002 | Functional  Testing | Query: "Provide suggestions to enhance vocabulary in my narrative essay." | Seasonal tips should be provided. | ✅ Passed | Tester 2 |
| TC-003 | Performance  Testing | API response time under  500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fix incorrect grammar suggestions for future tense in essay "The effects of pollution.". | Data accuracy should be improved. | ✅ Fixed | Develop er |
| TC-005 | Final Validation | Ensure UI displays correctly across all devices, including tablets and phones. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Tester 2 |
| TC-006 | Deployment  Testing | Host the WriteWise AI app using Streamlit or similar platform. | App should be accessible online. | 🚀 Deployed | DevOps |

## Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**