NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

GROUP PROJECT

Topic: Web Application with AI and/or Web3

I. Background

You are a web application developer at a regional bank looking to expand its digital footprint. You are to create a web application for a bank, addressing real-world banking challenges and user needs. The bank aims to enhance customer engagement, streamline banking operations, and offer innovative financial products. Your task is to propose a robust web application that addresses these goals and supports the bank's expansion strategy. (possible solution: 1. to provide users with a platform where they can use web3 technology include blockchain. 2. genAI technology, eg, generate personalized content, such as images, text, or music, using advanced generative AI technologies.)

Alternatively, you can play another roles in a big company and create a web application (you can specified both the role or your target audience, see part II) to solve a problem using AI/Web3. You may use real cases such as attached problem statement 1,2,3 and 4.

II. Suggested Project Topics

1. Customer Engagement: Develop features that improve the user experience, such as personalized financial advice, easy navigation, and interactive tools for financial planning.
2. Operational Efficiency: Implement functionalities that automate routine tasks, reduce processing times for transactions, and facilitate seamless communication between various bank departments.
3. Innovative Financial Products: Introduce new services such as digital wallets, peer-to-peer lending, and micro-investment platforms.
4. User Account Management: Secure login/logout
5. Customer Support: Live chat, AI-driven chatbot for 24/7 assistance, and a comprehensive FAQ section. The chatbot should smoothly transition the user to human customer support, providing a seamless service experience. The chatbot should have a system for collecting user feedback to continuously improve its performance and user satisfaction.
6. Dashboard and Reporting: Real-Time Financial Updates and Alert: The chatbot should provide real-time updates on relevant financial news, changes in stock prices, or significant account movements.
7. Usability: Ensure the web application is intuitive and accessible on various devices, including mobiles, tablets, and desktops.

Technologies to Use (Guideline):
- Frontend: HTML, CSS, JS or others
- Backend: Python Flask or others
- Database: SQLlite or MongoDB or others
- Cloud Services: render or AWS or others
- AI Model Selection: Choosing the right models for each type of content generation (e.g., GPT for text, DALL-E for images, VQ-VAE for music).
- API Integration: Using APIs to integrate various AI models and manage computational loads efficiently.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

III Suggested Structure of your report (below is only a guideline)

1. Project Proposal
- Objective: Clearly define what the project aims to achieve and how it incorporates generative AI. Identify a problem that your application will solve or a unique feature it will offer.
- **Scope**: Limit the scope to what can realistically be achieved within your timeframe and available resources.
2. Research and Literature Review
- Background Study: Research existing technologies and similar applications. This will help you understand the current state of technology and identify your project's unique value.
- Sources: Utilize academic journals, books, and reputable online resources. Cite all sources properly in your project documentation.
3. Technology and Tools
- Generative AI Models: Decide whether to use existing models like OpenAI's GPT or image generation models, or if you need to train a custom model.
- Development Tools: Choose appropriate programming languages, frameworks, and libraries that support AI integration, like Python with TensorFlow or PyTorch, and web frameworks like Flask.
4. Design and Development
- Prototyping: Create wireframes and prototypes to visualize the user interface.
- Development Phases: Break the project into manageable phases, each with its own goals and deadlines.
- Collaboration: Use version control systems like Git for collaborative coding, especially if working in a team.
5. Implementation
- Coding: Start coding the application, integrating the AI model with the web interface.
- Testing: Regularly test the application for bugs and ensure the AI model performs as expected under different scenarios.
6. Analysis and Evaluation
- Performance Metrics: Define metrics to evaluate the AI model's performance and the application's usability.
- User Testing: If possible, organize a session with potential users to gather feedback and observe user interaction.
7. Documentation
- Project Report: Prepare a comprehensive report that includes your research, design choices, development process, challenges faced, and outcomes.
8. Presentation
- Final Presentation: Prepare to showcase your project to your professors and classmates. This could include a live demonstration, a walkthrough of the code, and a discussion of the project's findings and potential future work.
9. Ethical Considerations and Compliance
- Ethical AI Use: Consider the ethical implications of your AI application, especially if it handles sensitive data.
- **Compliance**: Make sure your project complies with any university guidelines for ethical research and data privacy.
10. Reflection
- Self-Assessment: Reflect on what you learned through the project and what could be improved.
- Peer Feedback: Seek feedback from peers and mentors to gain different perspectives on your work.

IV. Project Deliverables:

1. A deck of presentation slides (10 to 15) that summarize your important findings and recommendations, including demo (if any)

2. (if any) The programming script and data used.

3. Submit to NTULearn.