Group Project Guidelines | Python Streamlit App

Project Objectives

- Reinforce your understanding of core Python and data processing.
- Introduce best practices in modular programming and user interface development.
- Provide hands-on experience with app design, version control, and presentation.
- Simulate real-world product building using data and logic.

Project Phases and Expectations

Phase 1: Setup (15 Marks)

- Create and activate a virtual Python environment.
- Define all required packages in a `requirements.txt` file.
- Ensure the environment is reproducible and easy for others to set up.

Phase 2: Development (40 Marks)

- Maintain a clear and logical code structure, including:
  - Main Streamlit app file
  - Utility modules (functions, classes)
  - Folder(s) for data and assets
- Implement the following elements:
  - Input widgets (file uploads, text inputs, dropdowns, sliders, etc.)
  - Output components (charts, tables, computed results)
  - At least one custom Python function
  - Basic use of Object-Oriented Programming principles
  - Clean, modular, and well-documented code

Phase 3: Testing (20 Marks)

- Test the app locally with various input scenarios.
- Validate that the application handles edge cases and invalid inputs.
- Ensure the logic performs correctly and consistently.

Phase 4: Delivery (20 Marks)

- Use Git for version control with a clear commit history.
- Prepare and submit the following deliverables:
  - Screenshots or a short demo video showcasing key features
  - Well-structured and commented code repository
  - A written summary explaining your approach and learnings

Documentation (5 Marks)

- Provide a `README.md` file including:
  - Project objective and summary
  - Description of features and functionality

    - Instructions to run the app locally
    - (Optional) Link to a deployed version or GitHub repository

Evaluation Criteria

Phase	Marks
Phase 1: Setup	15
Phase 2:Development	40
Phase 3: Testing	20
Phase 4: Delivery	20
Phase5:Documentation	5
Total	**100**

Evaluation Notes

- Setup: Proper environment and dependency management.
- Development: Functionality, interface quality, and modular coding practices.
- Testing: Comprehensive testing coverage and handling of various scenarios.
- Delivery: Version control, demo quality, and submission completeness.
- Documentation: Clarity, organization, and completeness of the README file.

Optional Enhancement Students may choose to integrate a Large Language Model (LLM) API to enhance their application. Groq Cloud offers a free API tier. Visit https://cloud.groq.com to obtain an API key and review available code samples.

Final Note This project is designed to consolidate your technical learning and encourage real-world thinking. Collaborative effort, clear documentation, and structured problem-solving will be key to building a successful application. Reach out for support as needed.