# **Python Capstone Project Proposal**

### **Project Title:**

*Text Based Adventure game*

### **Group Members:**

* Sushma
* Smita

### **Project Overview:**

**Objective**: Developing a text based adventure game to play on all platforms (iOS, MacOS, Linux, Android & Windows). Users can create username, password, save game progress, scores and highest scores, choose options to unfold the story.

**Core Features:**

1. Users can create new usernames, password, and play the game by clicking on options provided in the story, each option fork leading to a different storyline, all the endings and gameplays are stored, Implementing player actions and interactions
2. Managing, game state, control flow of the events in each storyline while handling user input

**Optional Features:**

1. **Database Integration:** Implement a database to store scores persistently, game progress associated with each username.
2. **User Interface:** Develop a user-friendly interface using Tkinter or HTML/CSS for sign up page, for game play and for viewing menu options and statistics.
3. **Notifications:** Add a feature to notify users of upcoming deadlines.
4. **Visualization**: Visualizing statistics on the game's highest scores for top 10.
5. API integration
6. Game deployment through Docker, using AWS cloud services

### **Technologies and Tools:**

* **Programming Language:** Python
* **Libraries:** Tkinter for UI, SQLite for database
* **Tools:** Figma for UI design, Git for version control
* **Libraries**: SpaCy, matplotlib, pandas etc

### **Project Timeline**

* **Week 4&5:** Brainstorming, designing, and project proposal submission & Implement core features (task creation, management, and viewing)
* **Week 6:** Implement core features/advanced features if possible ( text time delay functions/font change, text feature manipulation)
* **Week 7:**
  + API integration with an LLM (Large language model) to generate advanced text based responses,
  + And then save the responses to a database, usernames, scores (sqlite)
  + Assigning weights to each choice for score generation (gradient scores), if possible, else designated scores (Green - 50, yellow - 30, red - 5)
  + Managing database of all the game plays played on all the devices
* **Week 8:** Game deployment through docker using AWS
* **Week 9:** Finalize the project, implement any remaining features, submit to GitHub & prepare the presentation slides and submit the final project.

**Roles and Responsibilities:**

* **Smita:** Responsible for task creation and management features.
* **Sushma:** Handles UI design and implementation.
* **Smita:** Manages database integration and storage.
* **Sushma:** In charge of testing, debugging, and documentation.

### **Expected Challenges:**

We are new to this, we are expecting challenges in each step. Managing time efficiently to implement optional features might be difficult.

### **Success Criteria:**

The project will be considered successful if the core features are fully functional, the application is user-friendly, and at least one optional feature is implemented. Additionally, the application should be free of major bugs and meet the testing criteria.

### **References/Resources:**

* "Python Programming for Beginners" book
* W3Schools Tutorials
* SQLite documentation
* Online Tkinter tutorials
* [Game creation guide](https://developers.knowivate.com/@kheersagar/developing-a-text-based-adventure-game-with-python-a-comprehensive-guide)
* [Time delay1](https://realpython.com/python-sleep/), [time delay-geeksforgeeks](https://www.geeksforgeeks.org/how-to-add-time-delay-in-python/), [text/font manipulation](https://pypi.org/project/fonttools/)
* Sept 8:
  + API integration: [What is an API? A Beginner's Guide to APIs | Postman](https://www.postman.com/what-is-an-api/), [What is an API (Application Programming Interface)? | Definition from TechTarget](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API), [What is an API? | API definition | Cloudflare](https://www.cloudflare.com/learning/security/api/what-is-an-api/)
  + time delay text:
  + html/css reading