Sprint 4:

1. What are the goals for the sprint 4? Create a list (backlog) of user stories (at least for this sprint, corresponding to the milestone 4 features of the requirements document): A short description of functionality from the user's perspective:

As a user	I want to make sure there are no vulnerabilities in this application	So I know my personal information is secure
As a user	I want to make sure the application has no accessibility violations	So I can use this application effectively.

2. Technology choices (hardware and software) and tools for program development (design, implementation, testing, version control, etc).
    - Flask: Lightweight Python server with all essential features for this project.
    - SQLite: Lightweight database that can be used with the Flask server to store data.
    - JavaScript/HTML/CSS: Frontend interface for the application.

VULNERABILITIES

- IDENTIFIER: SET_TOTAL_RESERVATIONS
- TEST CASE:
    1. Vulnerability scanner inspects the Flask server and finds nothing.
    2. Vulnerability detects a vulnerability.
- PRECONDITIONS: LOGIN
- INPUT VALUES: Source code/packages
- EXECUTION STEPS: The developer will activate a vulnerability scanner for Flask at its root directory.
- OUTPUT VALUES: Whether or not a vulnerability is found and what it is.
- POSTCONDITIONS: If any vulnerabilities are found, the developer will fix them.

ACCESSIBILITY

- IDENTIFIER: ACCESSIBILITY
- TEST CASE:
    1. WAVE finds no vulnerabilities in the site.
    2. WAVE finds vulnerabilities in the site.
- PRECONDITIONS: Able to navigate to every page
- INPUT VALUES: Web page
- EXECUTION STEPS: The user visits every page one by one and uses WAVE to scan.

- OUTPUT VALUES: What violations are present and where or that no violations are found.
- POSTCONDITIONS: If any violations are found, the developer will fix them.

- Documentation level: what kinds of documentation do you plan to do?
  - UML Diagram to visualize goals and sprints.
  - User manual/Readme to show users how to use the application.