**Project StrongBox**

### **Project Outline**

1. **Project Requirements and Planning**
   * Define core features and functionality.
   * Decide on the tech stack and libraries.
2. **Core Features**
   * Secure storage for passwords.
   * Password generator.
   * GUI for desktop application.
   * Browser accessibility.
   * Clickable system tray icon.
3. **Tech Stack and Libraries**
   * **GUI Framework**: PySide6.
   * **Password Storage and Encryption**: SQLite with SQLCipher for encryption.
   * **Password Generation**: Python’s “secrets” and “string” modules.
   * **Web Server**: Flask for browser accessibility.
   * **System Tray Icon**: Pystray.
4. **Project Structure**
   * Backend (Database and Encryption)
   * Frontend (GUI and Browser Interface)
   * Integration and Deployment

### **Detailed Steps**

1. **Backend Development**
   * Set up an SQLite database with SQLCipher for encrypted storage.
   * Implement functions for storing, retrieving, and deleting passwords.
   * Create a secure password generation function.
2. **Frontend Development (GUI)**
   * Design and implement the GUI using PySide6.
   * Integrate the password management functions with the GUI.
3. **Browser Accessibility**
   * Set up a web server using Flask.
   * Create endpoints for accessing the password manager.
   * Implement authentication and session management.
4. **System Tray Icon**
   * Use Pystray to create a clickable system tray icon that launches the GUI or browser interface.
5. **Integration and Testing**
   * Ensure seamless integration between all components.
   * Conduct thorough testing for security and functionality.
6. **Deployment**
   * Package the application for distribution on different platforms (Windows, macOS, Linux).