Selleck Elliott  
Proposal 1

**Project Proposal: Bodybuilding Progress Tracker App**

**1. Project Title:** 🌟

Bodybuilding Progress Tracker App

**2. Introduction** 🌟

The Bodybuilding Progress Tracker App is designed to help users efficiently manage their bodybuilding workouts and track their progress over time. By pulling information from a comprehensive bodybuilding book and utilizing an API for calendar tracking, this app aims to provide a seamless experience for fitness enthusiasts to plan, execute, and review their workouts.

**3. Objectives** 🌟

* To create a user-friendly web application that assists users in tracking their bodybuilding workouts.
* To provide a robust platform for users to store, manage, and analyze their workout data.
* To integrate calendar functionalities for users to schedule and track their workout progress.
* To showcase the knowledge and skills acquired in C#, HTML, CSS, JavaScript, REST JSON, and SQL (MySQL).

**4. Project Description** 🌟

The Bodybuilding Progress Tracker App addresses the need for organized and effective workout tracking among bodybuilding enthusiasts. The app will feature workout plans, progress tracking, and calendar integration to help users maintain a consistent workout routine. Users can input their workout details, view their progress over time, and plan future workouts using the calendar feature.

Key Features:

* Progress Tracking: Users can log their workout sessions and track progress over time.
* Calendar Integration: Users can schedule their workouts and receive reminders.
* Database Management: Secure storage of user workout data with SQL (MySQL).
* User-Friendly Interface: Intuitive UI for easy navigation and use.

**5. Technologies and Tools** 🌟

* Languages: C# for backend development, HTML/CSS/JavaScript for frontend
* Frameworks:NET MAUI for app development
* Database: MySQL for data storage
* Tools: Visual Studio, Git for version control, API for calendar functionalities

**6. Features** 🌟

* Progress Tracking: Enables users to log workout details (e.g., exercises, sets, reps, weights) and track their progress.
* Calendar Integration: Provides a calendar interface for users to schedule workouts and receive notifications.
* Database Management: Ensures secure storage and retrieval of workout data using MySQL.
* User Interface: Features a clean and intuitive UI for easy navigation and user interaction.

**7. Implementation Plan** 🌟

1. Initial Setup: Set up the development environment and initialize the project repository.
2. Backend Development: Develop the backend using C# and .Net MAUI, focusing on workout data management and API integration.
3. Frontend Development: Create the user interface using HTML, CSS, and JavaScript, ensuring responsiveness and usability.
4. Database Integration: Design and implement the MySQL database for storing workout data.
5. API Integration: Integrate a calendar API for scheduling and tracking workouts.
6. Testing: Conduct unit, integration, and user acceptance testing to ensure functionality and performance.
7. Deployment: Deploy the app to a suitable hosting platform (e.g., Azure App Service).

Potential Challenges:

* Ensuring seamless integration of the calendar API.
* Managing data security and user privacy.
* Creating an intuitive and responsive user interface.

**8. User Interface (UI) Design**

* Wireframes and Mockups: Create wireframes and mockups using tools like Figma.
* Design Principles: Focus on simplicity, intuitiveness, and mobile responsiveness to enhance user experience.

**9. Testing Plan** 🌟

* Unit Testing: Test individual components of the backend logic using NUnit.
* Integration Testing: Ensure seamless integration between different components (e.g., backend, database, API).
* User Acceptance Testing: Gather feedback from real users to identify and fix usability issues.

**10. Deployment**

* Hosting: Deploy the app to major app stores (Google Play Store, Apple App Store).
* Server Requirements: Cloud-based server with .NET MAUI support and MySQL database.
* Scalability: Implement vertical and horizontal scaling options to handle increased user traffic.

**11. Expected Outcome** 🌟

* + A fully functional Bodybuilding Progress Tracker App accessible via web browsers.
  + Success criteria: Positive user feedback, increased user engagement, and improved workout management for users.
  + Be jacked in 6-9 months

**12. Conclusion** 🌟

The Bodybuilding Progress Tracker App aims to provide an effective tool for bodybuilding enthusiasts to manage and track their workouts. By leveraging the knowledge gained from the course, this project will demonstrate proficiency in web development, database management, and API integration.

**13. References**

* “Bodybuilding for Beginners” by Kyle Hunt
* "Enterprise application patterns using .NET MAUI" by Michael Stonis
* MySQL documentation