

IT2143 - Visual Programming

Group Project

Group C1

Fitness Tracker

Group Members :-

No.	Registration Number	Name
01.	2020/ICT/114	W. M. P. S. Wijekoon
02.	2020/ICT/101	A. R. Wijesuriya
03.	2020/ICT/090	K. M. I. I. Madhushani
04.	2020/ICT/003	S. Diluxshana
05.	2020/ICT/103	R. M. H. M. Ranasinghe
06.	2020/ICT/057	C. H. Hettiarachchi

Contents

1. Introduction
2. Objectives
3. Methodology
 - I. Requirements gathering
 - II. Tools and Technologies
4. Implementation
 - I. Interface design
 - II. Database
 - III. Output
5. Conclusion
6. References

01. Introduction

In those days people's health and wellness, technology plays a big role in empowering individuals to achieve their fitness aspirations. Two fundamental pillars in this journey are Progress Tracking and nutrition Plans, which serve as the cornerstone for maintaining motivation and driving consistent improvements. The ability to set clear and attainable fitness goals provides users with a roadmap for their wellness journey. Whether aiming to lose weight, build muscle, or improve overall endurance, goal setting establishes a tangible endpoint and serves as a source of inspiration. Welcome to a realm where fitness isn't just a routine but a personalized, data-driven journey towards well-being.

02. Objectives

- Goal Setting and Progress Tracking:
Enable users to set fitness goals and track their progress over time. This helps in motivating users to stay consistent and make improvements.
- Personalized Recommendations:
Offer personalized workout plans, nutrition advice, and suggestions based on user data and goals. This enhances user engagement and adherence to healthy habits.

03. Methodology

I. Requirements gathering

- **User Story Workshops:** Conduct workshops to gather user stories, capturing the needs and expectations of fitness enthusiasts. Discuss goal-setting preferences, progress tracking features, and the desired user experience.
- **Stakeholder Interviews:** Engage with fitness trainers, nutritionists, and potential end-users to understand diverse perspectives. Extract insights on ideal goal-setting mechanisms, progress metrics, and personalized recommendations.
- **Data Collection and Analysis:** Identify key data points for user profiling, such as fitness levels, dietary preferences, and health conditions. Analyze existing fitness data models to ensure seamless integration and accuracy.
- **User Interface (UI) Mockups and Prototypes:** Develop UI mockups and interactive prototypes showcasing goal-setting interfaces, progress dashboards, and areas for personalized recommendations. Gather feedback to refine design elements.
- **Functional Requirements:** Define functional specifications for goal-setting modules, progress tracking algorithms, and recommendation engines. Specify the logical flow of data, ensuring a smooth user experience.
- **C# Programming Language Integration:** Specify technical requirements for C# programming language, outlining its role in database connectivity, algorithm implementation, and user interface development. Ensure compatibility with existing systems.
- **Security and Privacy Protocols:** Establish robust security measures for user data protection. Implement encryption algorithms and access controls, adhering to industry standards and legal requirements.
- **Scalability and Performance:** Define scalability requirements to accommodate a growing user base. Establish performance benchmarks for seamless goal tracking and recommendation delivery, ensuring optimal user experience.
- **Testing Protocols:** Develop comprehensive testing protocols for goal-setting workflows, progress tracking accuracy, and the efficacy of personalized recommendations. Conduct rigorous testing phases to identify and rectify potential issues.
- **Feedback Loops:** Integrate user feedback loops to continuously refine and improve the system. Implement mechanisms for users to provide input on goal-setting effectiveness, progress tracking accuracy, and the relevance of personalized recommendations.

- **Documentation:** Create thorough documentation outlining the software's functionalities, APIs, and data structures. Ensure that the documentation is accessible to both developers and end-users.
- **Compliance and Regulations:** Ensure that the software adheres to relevant health and data protection regulations. Implement features that allow users to customize privacy settings and control the sharing of their fitness data.

II. Tools and Technologies

- **IDE: Visual Studio Code**
Utilize Visual Studio for C# development, providing a feature-rich IDE for coding, debugging, and testing. Leverage its integration with .NET frameworks for seamless C# development.
- **Programming Language: C#**
C# is the primary programming language for developing the backend logic and algorithms. Its object-oriented nature and integration with .NET make it well-suited for robust and scalable applications.
- **Version Control: GitHub**
Implement GitHub for version control to track changes in the codebase, facilitate collaboration among developers, and ensure a systematic approach to software development.
- **Prototype Design: Canva**
Canva as a user-friendly and versatile prototyping tool for designing mockups and wireframes of the software's user interface. Canva allows for the creation of visually appealing and interactive prototypes that can be easily shared and iterated upon.
- **Cloud Storage Service: Google Drive**
Implement GoogleDrive as a cloud storage solution to securely store and manage various data assets for the software. Utilize Google Drive's scalable infrastructure for seamless data access, storage, and collaboration.

04. Implementation

I. Interface design



Menu Page

Fitness Results Page

Nutrition Advice Page

II. Database

It's essential to note that the decision not to use a database is context-dependent and should be carefully considered based on the specific requirements and goals of the fitness software. While foregoing a traditional database simplifies certain aspects, it's crucial to ensure that the chosen data storage approach aligns with the software's intended functionality and future scalability needs.

III. Output

[FitMe Demo Video](#)

[ProjectFiles - GitHub](#)

05. Conclusion

In conclusion, the development of fitness software using C# involves a thoughtful selection of tools and technologies to create a robust and user-centric application. By leveraging a combination of programming languages, frameworks, cloud services, and design tools, we aim to provide a seamless and engaging fitness experience. The utilization of C# as the primary programming language, along with .NET Core for web development, ensures a scalable and high-performance application.

The decision to incorporate Google Drive for cloud storage enhances data accessibility, collaboration, and real-time synchronization, while Canva serves as a valuable prototyping tool for designing an intuitive and visually appealing user interface. These tools contribute to a holistic approach to fitness software development, addressing aspects of data storage, user interaction, and collaborative design.

By strategically choosing and integrating these tools, we aim to strike a balance between functionality and simplicity, facilitating agile development cycles, reducing maintenance overhead, and promoting user engagement. Whether through personalized goal setting, progress tracking, or tailored recommendations, the fitness software aims to empower users on their wellness journey.

In the ever-evolving landscape of fitness technology, this approach seeks to create a dynamic and adaptable solution, aligning with the diverse needs of users and providing a foundation for continuous improvement and innovation in the realm of health and well-being.

06. References

For AI generated ideas and other things: [ChatGPT](#)
For get ideas through the internet: [Google](#)