

In the following document the prompts used to generate and refine the README.md and their corresponding responses are displayed.

The use case description (1st prompt) is directly copied from the 3rd deliverable of the group project.

Model: ChatGPT 4o

1) Prompt:

“

In a readme for my MVC dotnet project describe the use case workflow. This is the use case in greek “Use Case Complete Capsule Σύνοψη Περιγραφή Η περίπτωση χρήσης “Complete Capsule” επιτρέπει στο χρήστη να δει το περιεχόμενο μιας Κάψουλας Γνώσεων και να απαντήσει στις αντίστοιχες ερωτήσεις Σωστού-Λάθους. Actors Student Βασικές Ροές Ενέργειες Δεδομένα Log In Ο χρήστης εφόσον έχει εγγραφεί στην πλατφόρμα (προαπαίτηση), συνδέεται με τα προσωπικά του στοιχεία (user credentials). User Credentials 1. Username 2. Email 3. Password Navigate To Capsule Page Ο χρήστης είτε μέσω του Home Page είτε μέσω του Find Page, μεταβαίνει στο Class Page της τάξης που τον ενδιαφέρει και επιλέγει το Capsule που θέλει να ξεκινήσει. Read Content and Watch Video Ο χρήστης βλέπει το Video που αντιστοιχεί στο Capsule και διαβάζει τις αντίστοιχες σημειώσεις Capsule Video URL, Capsule Content Question Answering Ο χρήστης απαντά τις ερωτήσεις Σωστού- Λάθους του Capsule. True False Question Redirection to next Capsule Εφόσον οι ερωτήσεις απαντηθούν σωστά, καταγράφεται η ολοκλήρωση του Capsule από το χρήστη και ο χρήστης ανακατευθύνεται στο επόμενο Capsule ή αν δεν υπάρχει στο Class Page. Η περίπτωση χρήσης τερματίζεται. Εναλλακτικές Ροές Wrong Answers ?? Στην περίπτωση που ο χρήστης απαντήσει” Add an overview section, an actor section and the rest of the sections described in the use case.

“

Response:

“

Here's the **Use Case Workflow** section in **Markdown** format for your **.NET project README**:

md

📖 Use Case: Complete Capsule

🎯 Overview

The **Complete Capsule** use case allows a **student** to:

- View the content of a **Capsule**.

- Watch an associated **Video**.
- Answer **True/False questions** related to the capsule.
- Successfully complete the capsule and move on to the next one.

👤 Actors

- **Student**: A registered user who interacts with Capsules.

🔑 Log In

- The user logs into the platform using their **credentials**.
- **Pre-requisite**: The user must be registered.

📋 **Required Data**

Field	Description	
-----	-----	
Username	The user's unique identifier.	
Email	The user's registered email.	
Password	The user's secure password.	

🔗 Navigate to Capsule Page

- The user can access a **Capsule** in two ways:
 - From the **Home Page**.

- From the **Find Page** → Navigating to a specific **Class Page**.
- The user selects the **Capsule** they want to start.

Read Content & Watch Video

- The user views the **Capsule's video** and reads the **provided notes**.

Capsule Data

Field	Description	
-----	-----	
Capsule Video URL	Link to the educational video.	
Capsule Content	Text-based learning material.	

Answer True/False Questions

- The user answers **True/False** questions related to the capsule.

Question Data

Field	Description	
-----	-----	
Question ID	Unique identifier of the question.	
Answer	True/False response from the user.	
CorrectAnswer	The predefined correct answer.	

Completion and Redirection

- If the user **correctly answers all questions**, their completion is recorded.
- They are **redirected to the next Capsule**.
- If no more Capsules are available, they return to the **Class Page**.

Completion Tracking

Field	Description
-----	-----
Capsule ID	The unique identifier of the capsule.
Student ID	The unique identifier of the student.
Completion	Boolean indicating completion status.




Alternative Flow: Wrong Answers

If the user provides **incorrect answers**:

- The system **notifies** them of incorrect responses.
- The user is prompted to **retry** the questions.
- The Capsule is **not marked as completed** until all answers are correct.

End of Use Case

The use case completes when:

-  The user has successfully answered all True/False questions.
-  The Capsule is marked as **completed**.
-  The user is redirected to the **next learning step**.

🚀 **Future Enhancements**

- Add **hint mechanisms** for incorrect answers.
- Provide **progress tracking** for completed Capsules.
- Introduce a **quiz retry limit**.

◆ Why This Format?

- ✓ **Clear & Structured:** Each step is easy to follow.
- ✓ **Tabular Data:** Helps organize key information efficiently.
- ✓ **Markdown-Friendly:** Readable in GitHub, GitLab, and documentation tools.

Would you like any additional refinements? 🚀 🔥

”

2) Prompt:

“

spell check this "Welcome to UniPath Capsule System, an innovative e-learning platform designed to enhance student engagement through interactive educational capsules. Built using Microsoft .NET Core MVC, this project brings a structured and immersive learning experience to users. This project implements the Complete Capsule use case of the full UniPath Project, allowing students to access educational capsules, engage with interactive content, and track their progress."

”

Response:

“

Here is the **spell-checked and refined version** of your text for better clarity and readability:

****"Welcome to the UniPath Capsule System, an innovative e-learning platform designed to enhance student engagement through interactive educational capsules. Built using Microsoft .NET Core MVC, this project provides a structured and immersive learning experience for users.**

This project implements the **Complete Capsule** use case of the full **UniPath Project**, enabling students to access educational capsules, engage with interactive content, and track their progress.***

✅ **Corrections & Improvements:**

- "Brings" → "Provides" (More natural and professional)
- "allowing" → "enabling" (Better flow)
- "to users" → "for users" (More grammatically correct in this context)

Let me know if you need further refinements! 🚀 🔥

”

3) Prompt:

“

now write a section for technical implementation for mvc in markdown for the readme, make it simple only add a small explanation for model, view, controller

”

Response:

“

Here is a **simplified "Technical Implementation (MVC)"** section for your **README.md** in **Markdown** format:

md

🛠️ Technical Implementation (MVC)

The **UniPath Capsule System** follows the **Model-View-Controller (MVC)** architecture, ensuring a clean separation of concerns and maintainability.

📌 **Model (Data & Business Logic)**

- Represents the **application's data** and handles **database interactions**.

- Uses **Entity Framework Core** to interact with the **SQL database**.

- Example models: `User`, `Capsule`, `TrueFalseQuestion`, `CapsuleCompletion`.

🎨 View (User Interface)

- Responsible for displaying **dynamic content** using **Razor Pages**.
- Uses **HTML, CSS, and Bootstrap** for styling.
- Example views: `Capsule Details`, `Quiz Questions`, `Class Page`.

🧠 Controller (Request Handling)

- Acts as a **bridge** between the **Model** and **View**.
- Handles **HTTP requests**, processes **business logic**, and returns **responses**.
- Example controllers: `CapsuleController`, `ClassController`, `AccountController`.

This structured **MVC approach** ensures that the system is **scalable, maintainable, and easy to extend**.

Why This Version?

- ✅ **Short & Clear** – Only essential details, no unnecessary code.
- ✅ **Beginner-Friendly** – Easy for new developers to understand.
- ✅ **Fits Well in README** – Doesn't overload users with too much technical detail.

Would you like to add a **small diagram** for better visualization? 🚀

”

4) Prompt:

“

spell check this Technical Implementation This project follows the MVC (Model-View-Controller) architecture: MVC Structure Model: Handles business logic and data interactions. View: Presents dynamic content to users (Razor views). Controller: Manages requests and responses. Current Implementation ✅ Video-based capsules with learning content. ✅ True/False quizzes for knowledge assessment.

”

Response:

“

Here is the **spell-checked and refined** version of your text:

Technical Implementation

This project follows the **MVC (Model-View-Controller)** architecture.

MVC Structure

- **Model:** Handles **business logic** and **data interactions**.
- **View:** Presents **dynamic content** to users (**Razor views**).
- **Controller:** Manages **requests and responses**.

Current Implementation

- ✓ **Video-based capsules** with learning content.
 - ✓ **True/False quizzes** for knowledge assessment.
-

Improvements & Fixes

- "Handles business logic and data interactions." → More **consistent and clear**.
- "Presents dynamic content to users (Razor views)." → Made **more readable**.
- **Formatting improved** → Added line breaks for **better readability**.

This version ensures clarity while maintaining a **professional tone**. 🚀 Let me know if you need further refinements! 🔥

”

5) Prompt:

“

only spell check this " 🎓 UniPath Capsule System - MVC ☀️ Project Overview Welcome to UniPath Capsule System, an innovative e-learning platform designed to enhance student engagement through interactive educational capsules. Built using Microsoft .NET Core MVC, this project brings a structured and immersive learning experience to users. This project implements the Complete Capsule use case of the full UniPath Project, allowing students to access educational capsules, engage with interactive content, and track their progress. 🎯 Project Goal The UniPath Capsule System is designed to deliver video-based learning modules with integrated quizzes to assess student comprehension. The system tracks user progress, enabling a seamless and effective learning journey. 💡 Key Features ✓ Video Learning Capsules – Engaging instructional videos for each topic.

✓ Interactive True/False Quizzes – Reinforce learning through assessments. 🛠️ Technical Implementation This project follows the MVC (Model-View-Controller) architecture: MVC Structure Model: Handles business logic and data interactions. View: Presents dynamic content to users (Razor views). Controller: Manages requests and responses. Current Implementation ✓ Video-based capsules with learning content. ✓ True/False quizzes for knowledge assessment. 📺 Complete Capsule Use Case Workflow The Complete Capsule use case allows a student to: View the content of a Capsule. Watch an associated educational video. Answer True/False questions related to the capsule. Successfully complete the capsule and move on to the next one. 👤 Actors Student: A registered user who interacts with Capsules. ⚙️ Workflow Steps 1. Log In The student logs into the platform using credentials. Pre-requisite: The student must be registered. 📄 Required Data Field Description Username The user's unique identifier. Email The user's registered email. Password The user's secure password. 2. Navigate to Capsule Page The student can access a Capsule in two ways: From the Login Page. From the Class Page → Navigating to a specific Capsule Page. 3. Read Content & Watch Video The student views the Capsule's video and reads the provided notes. 📄 Capsule Data Field Description Capsule Video URL Link to the educational video. Capsule Content Text-based learning material. 4. Answer True/False Questions The student answers True/False questions related to the capsule. 📄 Question Data Field Description Question ID Unique identifier of the question. Answer True/False response from the student. CorrectAnswer The correct answer for validation. 5. Completion and Redirection If the student answers all questions correctly, their completion is recorded. They are redirected to the next Capsule or, if none exists, back to the Class Page. If they answer incorrectly, they receive feedback and can retry. 📄 Completion Tracking Field Description Capsule ID The unique identifier of the capsule. Student ID The unique identifier of the student. Completion Boolean indicating completion status. 🔄 Alternative Flow: Wrong Answers If the student answers a question incorrectly: The system provides immediate feedback. The student can retry the questions. They can only progress after answering all questions correctly. How to Run the Project Follow these steps to set up and run the .NET MVC project on your local machine. Prerequisites Ensure you have .NET9 installed Actions Apply the following commands in the following order: git clone https://github.com/tasosger/UniPath-MVC.git cd UniPath-MVC dotnet restore dotnet ef dotnet build dotnet run access the app at http://localhost:5010/ Generative AI Usage This project utilizes Generative AI to enhance content creation in the following ways: Copywriting & Content Generation AI-assisted tools were used to generate marketing copy, error messages, and UI text to ensure clarity and engagement. Documentation Assistance Sections of this README, including the installation steps and feature descriptions, were refined with AI-powered suggestions. While AI played a role in content creation, all generated text was reviewed and edited for accuracy, clarity, and relevance." dont add anything just say if it is correct or not

”

Response:

“

Your text is **correct** with no spelling errors. ✓

