

# CAREER ADVISOR

A Multi-Agent System using Gemini, LangChain, LangGraph, FAISS & Gradio

MATINA PAPADAKOU

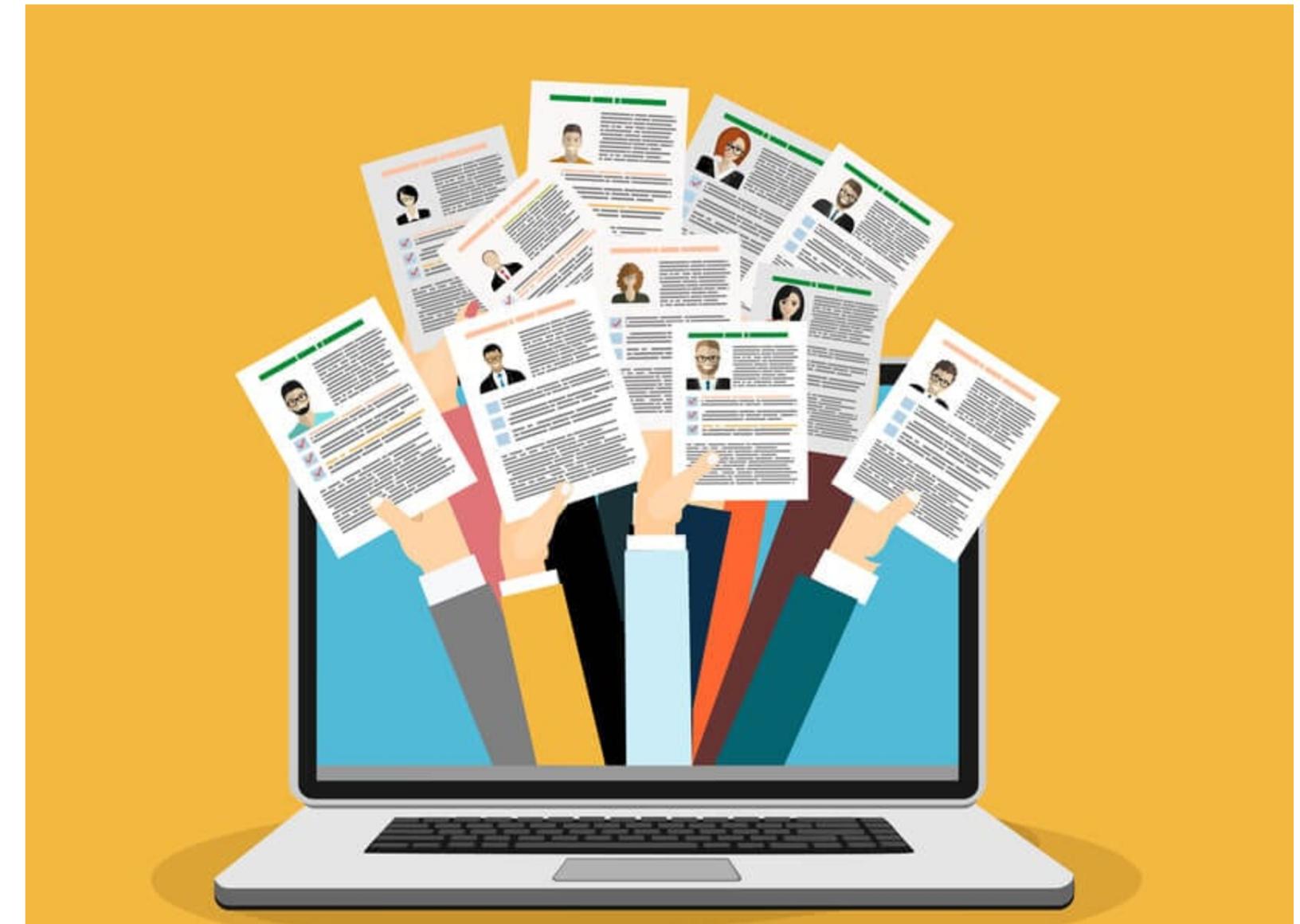
# WHY THIS PROJECT MATTERS

## Problem

- Hard to identify suitable careers
- Challenges with resumes & interviews

## Motivation

- Inspired by coursework in Intelligent Agents
- As a 4th-year student preparing to job hunt
- Created this chatbot assistant to support career preparation



# AI CAREER ADVISOR PIPELINE

This project provides an end-to-end AI advisor powered by multi-agent architecture:

-  [Profile Analyzer](#) – identifies skills, knowledge & interests
-  [Career Suggestion Agent](#) – recommends suitable career paths with role descriptions & required skills
-  [Learning Path Generator](#) – creates targeted study plans with courses & estimated learning times
-  [Resume Advisor \(RAG-powered\)](#) – analyzes resumes & provides feedback using FAISS + embeddings
-  [Mock Interview Agent](#) – simulates interviews with tailored questions & answers
-  [Personalized Learning Agent](#) – detects skill gaps & adapts learning plans to goals & preferences

# TECH STACK

## 1. Language Models & AI APIs

- Google Gemini (google.generativeai)
  - Model used: gemini-flash-latest
- LangChain (langchain\_core, langchain\_text\_splitters) – LLM orchestration, text splitting

## 2. Semantic Search & Embeddings

- HuggingFaceEmbeddings – Sentence embeddings for knowledge retrieval
  - Model used: sentence-transformers/all-MiniLM-L6-v2
- FAISS – Vector database for fast semantic search

## 3. Intelligent Agent Orchestration

- LangGraph (StateGraph) – Defines AI agent workflows and pipelines

## 4. PDF & Document Processing

- PyPDF2 – Extract text from PDFs
- unstructured.partition.pdf – Advanced PDF parsing
- Document (LangChain) – Wrap content for AI processing

# SYSTEM CAPABILITIES

## 1. Default Mode

- Analyze user profile
- Suggest suitable career roles
- Generate a general learning path

Ideal for users seeking quick career direction and study recommendations.

## 2. Interview Mode

- Provide resume feedback using RAG
- Simulate a mock interview

Perfect for users preparing for job applications and interviews.

## 3. Personalized Mode

- Analyze profile
- Create a personalized learning plan based on skill gaps and preferences

Best for users who want a custom roadmap aligned with their goals and learning style.

# 1. DEFAULT MODE

- **No role input: The system analyzes the profile and suggests two possible roles.**

The screenshot shows a dark-themed web application interface. At the top, there is a header section with a magnifying glass icon and the text "Προφίλ / Περιγραφή". Below it, a message states: "Έχω σπουδάσει πληροφορική. Ξέρω C#. Μόλις τελείωσα τις σπουδές μου, δεν έχω επαγγελματική εμπειρία, οπότε έχω γνώσεις Junior Developer." In the middle section, there is a heading "Επιλεγμένος Ρόλος (προαιρετικά)" followed by a dropdown menu. Below this, a file upload section shows a PDF named "MatinaPapadakouCV.pdf" with a size of "199.7 KB". At the bottom, there is a dropdown menu labeled "Λειτουργία" with the value "default" selected. A large grey button at the very bottom contains the text "Submit" with a rocket icon and "Clear" with a trash can icon.

# 1. DEFAULT MODE

- No role input: The system analyzes the profile and suggests two possible roles.**

**ΑΝΑΛΥΣΗ ΠΡΟΦΙΛ ΧΡΗΣΤΗ: Matina Papadakou (Junior Software Engineer)**

Επίπεδο Γνώσεων και Εμπειρίας

Το προφίλ αντιστοιχεί σε Junior Developer με τοχυρό ακαδημαϊκό υπόβαθρο στην Πληροφορική και πρακτική εμπειρία μέσω ακαδημαϊκών έργων, ηγεσίας σε φοιτητική πρωτοβουλία (UniAI) και εθελοντισμού. Η αποφοίτηση αναμένεται σύντομα (Σεπτέμβριος 2025). Ο στόχος είναι σαφώς ο ρόλος του Junior .NET Software Engineer.

**Βασικές Δεξιότητες και Γνώσεις (Κατηγοριοποιημένες)**

Κατηγορία	Δεξιότητα	Περιγραφή
Γλώσσες Προγραμματισμού & Frameworks	C# και .NET Core	Κεντρική δεξιότητα, με εξειδίκευση στο ASP.NET Core (MVC) και Entity Framework Core. Έχει εφαρμοστεί σε ολοκληρωμένο ακαδημαϊκό έργο (OnlineShopUniPi).
	Πολλαπλές Γλώσσες (Πολυγλωσσία Προγραμματισμού)	Γνώση σε Python, Java, C/C++, JavaScript, PHP, και Ruby. Αυτό υποδηλώνει ευρεία αντίληψη των αρχών προγραμματισμού.
	Front-End Development	Γνώση HTML, CSS και XML, απαραίτητη για την ανάπτυξη και συντήρηση ιστοσελίδων (εφαρμοσμένη εμπειρία στο UniAI και σε projects).
	Mobile Development (Java/Android)	Εμπειρία στην ανάπτυξη εφαρμογών για κινητά (Android Studio) χρησιμοποιώντας Java και XML, με ενσωμάτωση υπηρεσιών Firebase (PyLearn project).
Τεχνικά Εργαλεία & Αρχές	Object-Oriented Programming (OOP)	Ισχυρή θεωρητική και πρακτική βάση στις αρχές του Αντικειμενοστραφούς Προγραμματισμού, απαραίτητη για Software Engineering.

**1. ΡΟΛΟΣ: Junior Backend .NET Developer**

Αυτός ο ρόλος είναι η άμεση και φυσική εξέλιξη των ακαδημαϊκών και πρακτικών της γνώσεων, εστιάζοντας στον κώδικα και την αρχιτεκτονική εφαρμογών.

**Περιγραφή Ρόλου**

Ο Junior Backend .NET Developer είναι υπεύθυνος για τον σχεδιασμό, την ανάπτυξη, τη δοκιμή και τη συντήρηση των server-side λογικών (business logic) των εφαρμογών. Χρησιμοποιεί το .NET Core/C# για την κατασκευή ισχυρών και κλιμακούμενων RESTful APIs, αλληλεπιδρά με βάσεις δεδομένων μέσω ORM (Entity Framework Core) και συνεργάζεται στενά με τους Front-End Developers και τους DevOps Engineers.

**Απαραίτητες Δεξιότητες για τον Ρόλο**

Κατηγορία	Δεξιότητα	Σχέση με το Προφίλ της Matina
Προγραμματισμός	C#, .NET Core (ASP.NET Core MVC)	Κεντρική Δεξιότητα. Έχει εκτενή εμπειρία μέσω του OnlineShopUniPi project.
Αρχιτεκτονική	Object-Oriented Programming (OOP)	Ισχυρή Βάση. Η γνώση OOP είναι θεμελιώδης για τη συγγραφή ποιοτικού, συντηρήσιμου κώδικα.
Βάσεις Δεδομένων	SQL, Entity Framework Core	Εμπειρία στη χρήση ORM και SSMS για τη διαχείριση και την αλληλεπίδραση με δεδομένα.
Εργαλεία	Version Control (Git/GitHub)	Άριστη Χρήση. Απαραίτητη για τη συνεργατική ανάπτυξη κώδικα.
Soft Skills	Αναλυτική Σκέψη, Επίλυση Προβλημάτων	Βασικά χαρακτηριστικά που τονίζονται στην ανάλυση του προφίλ.

**Γιατί ταριάζει στη Matina Papadakou**

Αυτός ο ρόλος αξιοποιεί πλήρως την τεχνολογική της εξειδίκευση. Οι γνώσεις της στο ASP.NET Core (MVC) και το Entity Framework Core την καθιστούν άμεσα παραγωγική στη Microsoft Stack. Η εμπειρία της στη διαχείριση RESTful APIs και η πρακτική εφαρμογή σε μεγάλα ακαδημαϊκά έργα επιβεβαιώνουν την ικανότητά της να αναλάβει ρόλους backend. Επιπλέον, η κατοχή της πιστοποίησης Azure AI Fundamentals δίνει ένα προβάδισμα στην κατανόηση των σύγχρονων αρχιτεκτονικών Cloud.

**2. ΡΟΛΟΣ: Junior DevOps/Cloud Engineer**

# 1. DEFAULT MODE

- When a role is provided, the system analyzes the profile and generates a learning plan accordingly.

**ΑΝΑΛΥΣΗ ΠΡΟΦΙΛ ΧΡΗΣΤΗ: Matina Papadakou (Junior Software Engineer)**

Επίπεδο Γνώσεων και Εμπειρίας

Το προφίλ αντιστοιχεί σε Junior Developer με τοχυρό ακαδημαϊκό υπόβαθρο στην Πληροφορική και πρακτική εμπειρία μέσω ακαδημαϊκών έργων, ηγεσίας σε φοιτητική πρωτοβουλία (UniAI) και εθελοντισμού. Η αποφοίτηση αναμένεται σύντομα (Σεπτέμβριος 2025). Ο στόχος είναι σαφώς ο ρόλος του Junior .NET Software Engineer.

**Βασικές Δεξιότητες και Γνώσεις (Κατηγοριοποιημένες)**

Κατηγορία	Δεξιότητα	Περιγραφή
Γλώσσες Προγραμματισμού & Frameworks	C# και .NET Core	Κεντρική δεξιότητα, με εξειδίκευση στο ASP.NET Core (MVC) και Entity Framework Core. Έχει εφαρμοστεί σε ολοκληρωμένο ακαδημαϊκό έργο (OnlineShopUniPi).
	Πολλαπλές Γλώσσες (Πολυγλωσία Προγραμματισμού)	Γνώση σε Python, Java, C/C++, JavaScript, PHP, και Ruby. Αυτό υποδηλώνει ευρεία αντίληψη των αρχών προγραμματισμού.
	Front-End Development	Γνώση HTML, CSS και XML, απαραίτητη για την ανάπτυξη και συντήρηση ιστοσελίδων (εφαρμοσμένη εμπειρία στο UniAI και σε projects).
	Mobile Development (Java/Android)	Εμπειρία στην ανάπτυξη εφαρμογών για κινητά (Android Studio) χρησιμοποιώντας Java και XML, με ενσωμάτωση υπηρεσιών Firebase (PyLearn project).
Τεχνικά Εργαλεία & Αρχές	Object-Oriented Programming (OOP)	Ισχυρή θεωρητική και πρακτική βάση στις αρχές του Αντικειμενοστραφούς Προγραμματισμού, απαραίτητη για Software Engineering.

**1. ΡΟΛΟΣ: Junior Backend .NET Developer**

Αυτός ο ρόλος είναι η άμεση και φυσική εξέλιξη των ακαδημαϊκών και πρακτικών της γνώσεων, εστιάζοντας στον κώδικα και την αρχιτεκτονική εφαρμογών.

**Περιγραφή Ρόλου**

Ο Junior Backend .NET Developer είναι υπεύθυνος για τον σχεδιασμό, την ανάπτυξη, τη δοκιμή και τη συντήρηση των server-side λογικών (business logic) των εφαρμογών. Χρησιμοποιεί το .NET Core/C# για την κατασκευή ισχυρών και κλιμακούμενων RESTful APIs, αλληλεπιδρά με βάσεις δεδομένων μέσω ORM (Entity Framework Core) και συνεργάζεται στενά με τους Front-End Developers και τους DevOps Engineers.

**Απαραίτητες Δεξιότητες για τον Ρόλο**

Κατηγορία	Δεξιότητα	Σχέση με το Προφίλ της Matina
Προγραμματισμός	C#, .NET Core (ASP.NET Core MVC)	Κεντρική Δεξιότητα. Έχει εκτενή εμπειρία μέσω του OnlineShopUniPi project.
Αρχιτεκτονική	Object-Oriented Programming (OOP)	Ισχυρή Βάση. Η γνώση OOP είναι θεμελιώδης για τη συγγραφή ποιοτικού, συντηρήσιμου κώδικα.
Βάσεις Δεδομένων	SQL, Entity Framework Core	Εμπειρία στη χρήση ORM και SSMS για τη διαχείριση και την αλληλεπίδραση με δεδομένα.
Εργαλεία	Version Control (Git/GitHub)	Άριστη Χρήση. Απαραίτητη για τη συνεργατική ανάπτυξη κώδικα.
Soft Skills	Αναλυτική Σκέψη, Επίλυση Προβλημάτων	Βασικά χαρακτηριστικά που τονίζονται στην ανάλυση του προφίλ.

**Γιατί ταριάζει στη Matina Papadakou**

Αυτός ο ρόλος αξιοποιεί πλήρως την τεχνολογική της εξειδίκευση. Οι γνώσεις της στο ASP.NET Core (MVC) και το Entity Framework Core την καθιστούν άμεσα παραγωγική στη Microsoft Stack. Η εμπειρία της στη διαχείριση RESTful APIs και η πρακτική εφαρμογή σε μεγάλα ακαδημαϊκά έργα επιβεβαιώνουν την ικανότητά της να αναλάβει ρόλους backend. Επιπλέον, η κατοχή της πιστοποίησης Azure AI Fundamentals δίνει ένα προβάδισμα στην κατανόηση των σύγχρονων αρχιτεκτονικών Cloud.

**2. ΡΟΛΟΣ: Junior DevOps/Cloud Engineer**

# 1. DEFAULT MODE

- When a role is provided, the system analyzes the profile and generates a learning plan accordingly.

The screenshot shows a dark-themed web application interface. At the top, there is a header bar with a profile icon and the text "Προφίλ / Περιγραφή". Below the header, a message states: "Έχω σπουδάσει πληροφορική, Ξέρω C#. Μόλις τελείωσα τις σπουδές μου, δεν έχω επαγγελματική εμπειρία, απότελε όμως γνώσεις Junior Developer." The main form area contains several input fields:

- A dropdown menu labeled "Επιλεγμένος Ρόλος (προαιρετικά)" with the value "Junior Backend .NET Developer".
- A file upload field showing a PDF named "MatinaPapadakouCV.pdf" with a size of "199.7 KB".
- A dropdown menu labeled "Λεπτουργία" with the value "default".

At the bottom of the form are two buttons: "Submit" and "Clear".

# 1. DEFAULT MODE

- When a role is provided, the system analyzes the profile and generates a learning plan accordingly.

Πλάνο Εκμάθησης: Junior Backend .NET Developer

Φάση 0: Προετοιμασία και Εργαλεία (1 Εβδομάδα)

Θέμα	Έννοιες προς Εκμάθηση	Εκτιμώμενος Χρόνος	Πόροι / Μαθήματα
Εγκατάσταση	Εγκατάσταση .NET SDK, Visual Studio (ή VS Code + C# Extension)	1 Ημέρα	Official .NET Documentation
Git & Version Control	Βασικές εντολές (commit, push, pull, branch), Workflow με GitHub/GitLab	3 Ημέρες	Pro Git Book (Δωρεάν online), YouTube: "Git Crash Course"
Terminal/CLI	Βασικές εντολές τερματικού (dotnet new, dotnet build, dotnet run)	3 Ημέρες	Microsoft Learn: Εισαγωγή στο .NET CLI

Φάση 1: Core C# και Αντικειμενοστραφής Προγραμματισμός (OOP) (4.5 Εβδομάδες)

Αυτή η φάση είναι η βάση. Η κατανόηση του C# και των αρχών του OOP είναι απαραίτητη πριν προχωρήσουμε στο web framework.

Θέμα	Έννοιες προς Εκμάθηση	Εκτιμώμενος Χρόνος	Πόροι / Μαθήματα
C# Fundamentals	Τύποι δεδομένων, Μεταβλητές, Τελεστές, Δομές ελέγχου (if/else, loops)	1.5 Εβδομάδες	Microsoft Learn: Get started with C#, YouTube: Mosh Hamedani - C# Beginner Course
OOP Βασικές Αρχές	Classes, Objects, Constructors, Properties, Access Modifiers, Structs	1.5 Εβδομάδες	Coursera/Udemy: Οποιοδήποτε εξειδικευμένο μάθημα OOP
Advanced C#	Interfaces, Abstract Classes, Generics, Delegates, Events, Exception Handling	1 Εβδομάδα	YouTube: Tim Corey - Advanced C# Topics
LINQ	Σύνταξη και χρήση του Language Integrated Query για συλλογές	3 Ημέρες	Microsoft Learn: Εισαγωγή στο LINQ

Φάση 2: Ανάπτυξη Web API με ASP.NET Core (4 Εβδομάδες)

Εδώ ξεκινά η ανάπτυξη του backend. Η εστίαση πρέπει να είναι στη δημιουργία RESTful υπηρεσιών.

Θέμα	Έννοιες προς Εκμάθηση	Εκτιμώμενος Χρόνος	Πόροι / Μαθήματα
ASP.NET Core Βασικά	Tι είναι το Kestrel, Middleware, Startup process, Configuration (.json)	1 Εβδομάδα	Microsoft Learn: Build web apps with ASP.NET Core
RESTful API	HTTP Methods (GET, POST, PUT, DELETE), Status Codes, Routing, Model Binding	1.5 Εβδομάδες	YouTube: Nick Chapsas (Εξειδικευμένα videos για API)
Dependency Injection (DI)	Service Lifecycles (Transient, Scoped, Singleton), Καθαρή αρχιτεκτονική με DI	1 Εβδομάδα	Microsoft Learn, YouTube: DI in .NET Core explained
Data Transfer Objects (DTOs)	Χρήση DTOs για διαχωρισμό του domain model από τα API requests/responses.	3 Ημέρες	Εφαρμογή σε μικρό project

# 2. INTERVIEW MODE

Προφίλ / Περιγραφή

Έχω σπουδάσει πληροφορική. Ξέρω C#. Μόλις τελείωσα τις σπουδές μου, δεν έχω επαγγελματική εμπειρία, οπότε έχω γνώσεις Junior Developer.

Επιλεγμένος Ρόλος (προαιρετικά)

Junior Backend .NET Developer

Βιογραφικό (PDF)

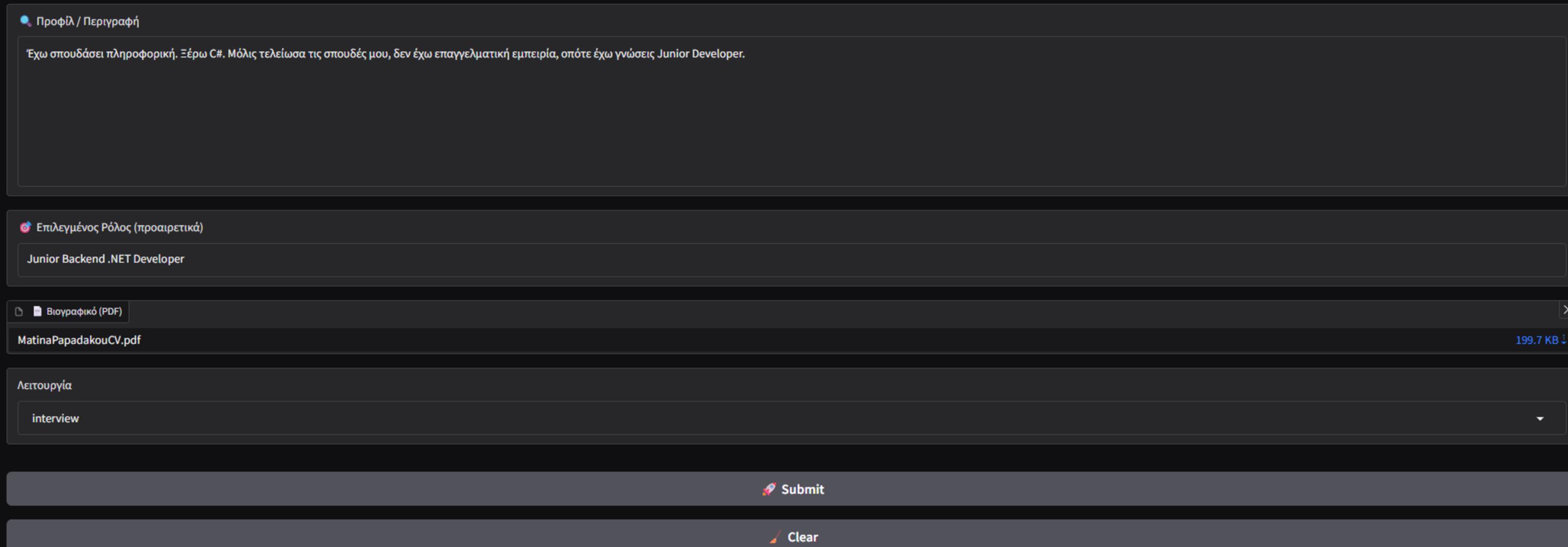
MatinaPapadakouCV.pdf 199.7 KB ↓

Λειτουργία

interview

Submit

Clear



# 2. INTERVIEW MODE

## • Resume Review

 **Σχόλια για το Βιογραφικό**

Η αξιολόγηση του βιογραφικού της Matina Papadakou δείχνει ότι πρόκειται για έναν πολύ ισχυρό υποψήφιο Junior Software Engineer με σημαντική ακαδημαϊκή επίδοση (υψηλό GPA) και αξέλογη πρακτική εμπειρία (UniAI, .NET project).

Το βιογραφικό είναι καλά δομημένο, επαγγελματικό και στοχευμένο στον επιθυμητό ρόλο (.NET Software Engineer). Οι προτεινόμενες βελτιώσεις εστιάζουν στην ενίσχυση της τεχνικής στόχευσης (.NET), στην ενσωμάτωση περισσότερων μετρήσιμων αποτελεσμάτων (Impact) και στη βελτίωση της αναγνωσιμότητας από συστήματα ATS (Application Tracking Systems).

---

**1. Γενικές Παρατηρήσεις και Δομή**

Στοιχείο	Παρατήρηση	Πρόταση Βελτίωσης
Στόχευση	Πολύ σαφής στόχευση (.NET), υποστηριζόμενη από το πρώτο project και τις δεξιότητες.	Διατήρηση της στόχευσης, ενίσχυση της προβολής των C#/NET δεξιοτήτων σε όλα τα τμήματα.
Ακαδημαϊκά	Το υψηλό GPA (8.65) είναι μεγάλο πλεονέκτημα για junior ρόλο.	Άριστη καταγραφή.
Έκταση	Αν και δεν αναφέρεται ρητά η έκταση, η ποσότητα πληροφορίας είναι μεγάλη. Για junior ρόλο, στόχος πρέπει να είναι η μία (1) σελίδα, αν και δύο (2) σελίδες είναι αποδεκτές λόγω της λεπτομερούς εμπειρίας.	Βεβαιωθείτε ότι η διάταξη είναι συμπεισμένη για να χωρέσει σε μία σελίδα, εάν είναι δυνατόν, χωρίς να χάσετε σημαντικές πληροφορίες.

### 1. OnlineShopUniPi (Κύριο .NET project)

- Εστιάστε στο .NET Core και την αρχιτεκτονική.

#### Προτεινόμενη Αναθεώρηση:

OnlineShopUniPi - C#, ASP.NET MVC, Entity Framework, SQL Server (SSMS) (GitHub)  
Academic project implementing a simplified online marketplace inspired by platforms like Vinted.

- Developed a full-stack application using \*\*ASP.NET MVC and Entity Framework\*\* for persistent data management in SQL Server.
- Implemented robust authentication/authorization features, supporting three distinct user roles (User, Guest, Admin).
- Designed and implemented core features including product management, shopping cart functionality, and order handling.
- Included a basic recommendation engine, suggesting products based on user purchase history and preferences.

# 2. INTERVIEW MODE

## • Mock Interview

### MOCK ΣΥΝΕΝΤΕΥΞΗ: JUNIOR BACKEND .NET DEVELOPER

Stage 1: Εισαγωγή, Κίνητρα και Προσωπικότητα (Behavioral & Soft Skills)

1. Ερώτηση: Πείτε μας λίγα λόγια για τον εαυτό σας και πώς οι σπουδές και τα έργα σας σας έχουν προετοιμάσει για έναν ρόλο Junior Backend Developer.

Απάντηση: Είμαι η Ματίνα, φοιτήτρια Πληροφορικής στο Πανεπιστήμιο Πειραιώς, με αναμενόμενη αποφοίτηση τον Σεπτέμβριο του 2025. Ενώ έχω εκτεθεί σε πολλές γλώσσες (Java, Python), έχω εστιάσει στρατηγικά στο οικοσύστημα της C# και του .NET Core.

Τα πιο σημαντικά για τον ρόλο είναι το ακαδημαϊκό έργο μου, το OnlineShopUniPi, όπου εφάρμοσα το ASP.NET Core MVC, Entity Framework Core και SQL Server. Εκεί ανέπτυξα όλη την αρχιτεκτονική του backend, από τη διαχείριση της βάσης δεδομένων και τη λογική των παραγγελιών, μέχρι την υλοποίηση ρόλων χρηστών (authentication/authorization). Εππλέον, ο ρόλος μου ως IT Lead στην UniAI, μου έδωσε πρακτική εμπειρία στη διαχείριση τεχνικής υποδομής (VPS setup) και τον συντονισμό τεχνικών προσπαθειών, δεξιότητες που μεταφράζονται άμεσα στην ομαδική εργασία και την επίλυση προβλημάτων.

2. Ερώτηση: Στο βιογραφικό σας αναφέρετε γνώσεις σε πολλές γλώσσες (Python, Java, C++). Τι ήταν αυτό που σας έκανε να εστιάσετε συγκεκριμένα στη C# και να επιδιώκετε έναν ρόλο Backend .NET;

Απάντηση: Αν και η έκθεση σε πολλές γλώσσες ήταν πολύτιμη για την κατανόηση διαφορετικών paradigms, η C# και το .NET Core με κέρδισαν για δύο κύριους λόγους: Πρώτον, λόγω της ισχυρής τυποποίησης και της καθαρότητας του κώδικα, κάτι που είναι κρίσιμο για τη συντήρηση μεγάλων εφαρμογών στο backend. Δεύτερον, το ASP.NET Core προσφέρει ένα εξαιρετικά αποδοτικό και ευέλικτο framework για τη δημιουργία RESTful APIs και web services, κάτι που με ενδιαφέρει ιδιαίτερα. Το project OnlineShopUniPi μου έδειξε πόσο γρήγορα και μεθοδικά μπορούσα να αναπτύξω μια ισχυρή backend λογική χρησιμοποιώντας αυτό το stack.

6. Ερώτηση: Στο Entity Framework Core, εξηγήστε τη διαφορά μεταξύ `IQueryable` και `IEnumerable` όσον αφορά την εκτέλεση ερωτημάτων στη βάση δεδομένων.

Απάντηση: Αυτή είναι μια κρίσιμη διαφορά για την απόδοση του backend:

Χαρακτηριστικό	<code>IQueryable</code>	<code>IEnumerable</code>
Πού γίνεται το φίλτραρισμα;	Στη βάση δεδομένων (Server-side)	Στη μνήμη της εφαρμογής (Client-side)
Πώς εκτελείται;	Εκτελείται μόνο όταν γίνει απαραίτητο (deferred execution) και μπορεί να τροποποιηθεί (π.χ., με <code>Where</code> ή <code>Select</code> ) πριν την εκτέλεση.	Εκτελείται αμέσως ή φέρνει όλα τα δεδομένα στη μνήμη, και μετά φίλτράρει.
Επίπτωση στην Απόδοση	Πολύ καλύτερη απόδοση. Στέλνει μόνο τον απαραίτητο SQL κώδικα στη DB.	Κακή απόδοση για μεγάλες συλλογές, καθώς μεταφέρει περιττά δεδομένα από τη DB στην εφαρμογή.

Συμπέρασμα: Ως Backend Developer, πρέπει να προσπαθούμε να χρησιμοποιούμε `IQueryable` (από μεθόδους όπως το `DbSet<T>.Where`) για φίλτραρισμα και pagination, ώστε να εκτελείται η λογική στον SQL Server και να μειώνεται η κίνηση δεδομένων.

# 3. PERSONALIZED MODE

Προφίλ / Περιγραφή

'Έχω σπουδάσει πληροφορική. Ξέρω C#. Μόλις τελείωσα τις σπουδές μου, δεν έχω επαγγελματική εμπειρία, οπότε έχω γνώσεις Junior Developer.

Επιλεγμένος Ρόλος (προαιρετικά)

Junior Backend .NET Developer

Βιογραφικό (PDF)

MatinaPapadakouCV.pdf 199.7 KB

Λειτουργία

personalized

Στυλ Μάθησης

visual

Στόχοι Καριέρας

Να γίνω Senior Backend Developer σε 4 χρόνια.

Submit

Clear

# 3. PERSONALIZED MODE

 **Ρόλος:** Junior Backend .NET Developer

 **Ανάλυση Gaps**

Η Matina Papadakou εμφανίζει ένα εξαιρετικά ισχυρό προφίλ για τη θέση του Junior Backend .NET Developer. Έχει καλύψει σχεδόν όλα τα must-have τεχνικά και ήπια κριτήρια που απαιτούνται, με ένα επίπεδο γνώσης που την τοποθετεί στην ανώτερη κλίμακα των Junior υποψηφίων.

Τα κενά που εντοπίζονται είναι τυπικά για έναν υποψήφιο που προέρχεται από ακαδημαϊκό περιβάλλον και επικεντρώνονται κυρίως σε βιομηχανικές βέλτιστες πρακτικές (best practices) και εργαλεία ανάπτυξης/υποδομών.

---

### Συγκριτική Ανάλυση & Καταγραφή Gaps

Ακολουθεί η ανάλυση των δεξιοτήτων σε σχέση με τις απαίτησεις του ρόλου:

**1. Δεξιότητες που Λείπουν (Missing Skills - Gaps)**

Αυτές είναι οι δεξιότητες που δεν αναφέρονται καθόλου στο προφίλ ή θεωρούνται κρίσιμες για τη σύγχρονη βιομηχανία, αλλά δεν έχουν αναδειχθεί από το ακαδημαϊκό έργο.

Κατηγορία	Απαιτούμενη Δεξιότητα	Σημασία για το Ρόλο
Δοκιμές (Testing)	Unit Testing Frameworks (π.χ., xUnit, NUnit)	Απαραίτητο για τη διασφάλιση της ποιότητας. Ένας Junior καλείται να γράφει απλά Unit Tests για τον κώδικα του.
Υποδομή (DevOps)	Containers (Docker/Kubernetes)	Βασική εξοικείωση με το πώς συσκευάζονται και αναπτύσσονται οι εφαρμογές .NET σήμερα. Αυτή η γνώση είναι πλέον de facto standard.
Αρχιτεκτονική	SOLID Principles	Αν και έχει ισχυρή βάση σε OOP, δεν αναφέρεται ρητά η γνώση των αρχών SOLID, οι οποίες είναι θεμελιώδεις για καθαρό και επεκτάσιμο κώδικα σε C#.
Αρχιτεκτονική	Συγκεκριμένα Design Patterns (π.χ., Repository Pattern, Dependency Injection)	Χρειάζεται να μεταφράσει τη θεωρητική γνώση (UML/OOP) στην πράξη, υιοθετώντας τα καθιερωμένα patterns που χρησιμοποιούνται σε web εφαρμογές .NET.
Αποσυγχρονισμός	Queuing Systems (π.χ., RabbitMQ, Azure Service Bus)	Δεν αναφέρεται. Αν και είναι "nice-to-have", σε πολλές σύγχρονες εφαρμογές .NET, η βασική γνώση asynchronous messaging είναι χρήσιμη.

# 3. PERSONALIZED MODE

## 2. Σε Ποιες Δεξιότητες είναι Αρχάριος/Πρώιμου Σταδίου (Early/Junior Stage)

Στις παρακάτω δεξιότητες, η υποψήφια έχει τη θεωρητική γνώση και βασική εμπειρία, αλλά θα χρειαστεί επίβλεψη και καθοδήγηση για να φτάσει στο επίπεδο των βιομηχανικών προτύπων.

Κατηγορία	Δεξιότητα	Επίπεδο Γνώσης/Εμπειρίας	Ανάγκη Καθοδήγησης
Web Development	ASP.NET Core (API Focus)	Έχει εμπειρία σε ASP.NET Core, αλλά κυρίως με μοντέλο MVC. Χρειάζεται εξάσκηση στη δημιουργία καθαρών, production-ready RESTful Web APIs και χρήση εργαλείων όπως το Swagger/OpenAPI.	
Βάσεις Δεδομένων	Βελτιστοποίηση SQL	Γνωρίζει SQL (CRUD/EF Core), αλλά ως Junior, θα είναι αρχάριος στη συγγραφή πολύπλοκων joins, βελτιστοποίηση ερωτημάτων (query optimization) και δημιουργία αποθηκευμένων διαδικασιών (Stored Procedures) υψηλής απόδοσης.	
Cloud Computing	Azure AI Fundamentals	Έχει την πιστοποίηση "Fundamentals" (γνώση εννοιών), αλλά όχι πρακτική εμπειρία στην ανάπτυξη (Deployment) και διαχείριση (Monitoring) μιας εφαρμογής σε Azure App Services ή Azure SQL.	
Διαχείριση Κώδικα	Code Review Standards	Γνωρίζει άριστα το Git/GitHub, αλλά η εφαρμογή των υψηλών προτύπων ποιότητας κώδικα και η ενεργή συμμετοχή σε αποτελεσματικά Code Reviews είναι διαδικασίες που αποκτώνται μόνο εντός οργανισμού.	

## 3. Σημαντικότερα Gaps για Κάλυψη (Critical Gaps to Address)

Για να μεταβεί η υποψήφια ομαλά από το ακαδημαϊκό στο εργασιακό περιβάλλον και να γίνει πλήρως παραγωγική, πρέπει να δώσει έμφαση στα εξής τεχνολογικά κενά:

- Unit Testing Πρακτικές:** Η άμεση εκμάθηση και εφαρμογή ενός Unit Testing Framework (.NET NUnit/xUnit) είναι απαραίτητη. Η ικανότητα να γράφει δοκιμές μονάδας και να κατανοεί την Test-Driven Development (TDD) φιλοσοφία είναι κρίσιμη για την ποιότητα του Backend κώδικα.
- Containerization (Docker):** Η έλλειψη γνώσης σε Docker μπορεί να καθυστερήσει τη συμμετοχή της σε σύγχρονες διαδικασίες DevOps. Η κατανόηση του πώς δημιουργείται ένα Dockerfile για μια εφαρμογή .NET και πώς εκτελείται σε containers είναι πλέον βασική απαίτηση.
- Design Patterns και SOLID:** Αν και έχει ισχυρό υπόβαθρο, πρέπει να επικεντρωθεί στο πώς οι αρχές SOLID, το Dependency Injection (χρησιμοποιείται εκτενώς στο ASP.NET Core) και patterns όπως το Repository Pattern, εφαρμόζονται στην πράξη.

# 3. PERSONALIZED MODE

## Πλάνο Επιτάχυνσης & Κάλυψης Gaps (12 Εβδομάδες)

Το πλάνο είναι χωρισμένο σε τρία βασικά modules που καλύπτουν τις πιο κρίσιμες ελλείψεις. Η συνολική διάρκεια είναι περίπου 12 εβδομάδες, υποθέτοντας 10-15 ώρες μελέτης ανά εβδομάδα.

### Module 1: Θεμελιώδης Ποιότητα Κώδικα & Δοκιμές (Testing)

(Προτεινόμενη Διάρκεια: 3 Εβδομάδες)

Αυτή είναι η πιο κρίσιμη ενότητα, καθώς ένας Senior Developer βασίζεται σε αρχές καθαρού κώδικα (Clean Code) και εξασφαλίζει τη λειτουργικότητα μέσω δοκιμών (Testing).

Τεχνολογία/Έννοια	Περιγραφή/Gaps που καλύπτονται
SOLID Principles	Οι θεμελιώδεις αρχές για το γράφιμο επεκτάσιμου, συντροήσιμου κώδικα σε C#. Πρέπει να τα δει στην πράξη.
Dependency Injection (DI)	Πρακτική εφαρμογή του Inversion of Control (IoC), το οποίο είναι ο πυρήνας του ASP.NET Core.
Unit Testing Frameworks	Εκμάθηση του xUnit (πιο διαδεδομένο) ή NUnit.
Mocking Frameworks	Εκμάθηση του Moq για τη δημιουργία ψεύτικων εξαρτήσεων (dependencies) και απομόνωση των μονάδων προς δοκιμή.
Clean Architecture Basics	Κατανόηση των επιπέδων (Layers) της εφαρμογής (Presentation, Application/Business Logic, Infrastructure) για καλύτερη οργάνωση.

#### 💡 Προτεινόμενα Visual Resources

Πόρος	Περιγραφή
YouTube Playlist (Tim Corey)	"C# SOLID Principles Explained" (Σειρά βίντεο). Ο Tim Corey είναι εξαιρετικός για οπτικούς μαθητές.
Coursera / Udemy	"The Ultimate ASP.NET Core Web API course" (Ανάλυση του τρόπου χρήσης DI και Patterns).
freeCodeCamp (YouTube)	Βρείτε ένα project-based tutorial για Unit Testing σε .NET. Η Matina χρειάζεται να γράψει tests για παλιό της κώδικα (legacy code).
Visual Tip:	Δημιουργήστε έναν απλό Calculator (C# Console App) και εξασκηθείτε να γράφετε Unit Tests για κάθε μέθοδο (Πρόσθεση, Αφαίρεση κ.λπ.) χρησιμοποιώντας xUnit και Moq.

# 3. PERSONALIZED MODE

## Module 4: Προετοιμασία για Mid-Level & Scaling (Scaling Preparation)

(Προτεινόμενη Διάρκεια: 2 Εβδομάδες + Συνεχής Εξάσκηση)

Αυτό το Module είναι προαιρετικό για την εύρεση της πρώτης Junior δουλειάς, αλλά είναι απαραίτητο για το στόχο των 4 ετών (Senior). Δείχνει ότι η Matina σκέφτεται πέρα από τη μονολιθική εφαρμογή.

Τεχνολογία/Έννοια	Περιγραφή/Gaps που καλύπτονται
Asynchronous Messaging	Θεωρητική κατανόηση του τι είναι οι Message Queues.
RabbitMQ/Kafka (Basics)	Πώς λειτουργούν τα publish/subscribe μοντέλα και πώς θα τα χρησιμοποιούσατε σε ένα .NET API (π.χ., για αποστολή email).
MassTransit/Rebus	Εξοικείωση με ένα .NET library που διευκολύνει την επικοινωνία με Queuing Systems.
Authentication/Authorization	Εμβάθυνση στα πρότυπα JWT (Tokens) και OAuth 2.0 (πώς υλοποιούνται στο ASP.NET Core).

### 💡 Προτεινόμενα Visual Resources

Πόρος	Περιγραφή
YouTube (Tim Corey/Nick Chapsas)	Συγκεκριμένα βίντεο για την ενσωμάτωση του RabbitMQ/MassTransit σε εφαρμογές .NET.
Pluralsight (ή Coursera)	Μαθήματα εστιασμένα στην Ασφάλεια των APIs και τη χρήση JWT (αν υπάρχει πρόσβαση).
Visual Tip:	Επεκτείνετε το Capstone Project: 'Όταν ένας χρήστης εγγράφεται (μέσω API), βάλτε το API να στέλνει ένα μήνυμα στο RabbitMQ και ένας άλλος Background Service να το διαβάζει και να εμφανίζει ένα log μήνυμα.

# 3. PERSONALIZED MODE

## Πλάνο 4ετίας: Από Junior σε Senior

Η γρήγορη κάλυψη των gaps δίνει στην Matina ένα σοβαρό πλεονέκτημα. Για να γίνει Senior σε 4 χρόνια, πρέπει να επικεντρωθεί σε δύο πυλώνες: Βάθος (Depth) και Εύρος (Breadth/Leadership).

Χρονοδιάγραμμα	Επίπεδο/Ρόλος	Βασικοί Στόχοι Εξέλιξης
Έτος 1	Junior Developer	Focus: Productive Code Writer. Να είναι ικανή να υλοποιεί User Stories (tasks) με ελάχιστη επίβλεψη. Άριστη γνώση SOLID, Testing, και REST API Design.
Έτος 2	Junior/Mid-Level Transition	Focus: Quality & Optimization. Ενεργή συμμετοχή σε Code Reviews, επίλυση πολύπλοκων bugs, βελτιστοποίηση SQL Queries, εμβάθυνση σε Cloud Services (Azure App Services, Functions, Key Vault). Κατανόηση Microservices αρχών.
Έτος 3	Mid-Level Developer	Focus: Ownership & Architecture. Να σχεδιάζει (design) νέα, μεσαίου μεγέθους features. Εξοικείωση με Distributed Systems (Queues, Caching – Redis). Επίδειξη καθοδήγησης (mentoring) σε νέους Juniors.
Έτος 4	Senior Developer	Focus: Strategic Thinking & Mentorship. Να αναλαμβάνει την αρχιτεκτονική μεγάλων εφαρμογών. Λήψη αποφάσεων για την επιλογή τεχνολογιών (Tech Stack Decisions). Ικανότητα διάγνωσης και επίλυσης προβλημάτων παραγωγής (Production issues) σε πραγματικό χρόνο.

### 🔑 Συμβουλή για το Visual Learning

Για να επιταχύνει τη διαδικασία, η Matina πρέπει να μετατρέπει τις σημειώσεις της σε οπτικά διαγράμματα:

- Διαγράμματα Ροής (Flowcharts): Σχεδιάστε τη ροή ενός request (από τον χρήστη στο API, μέσω του Controller, του Service Layer, του Repository και πίσω).
- UML/Σχηματικές Αναπαραστάσεις: Χρησιμοποιήστε διαγράμματα για να δείξετε πώς συνδέονται τα components (π.χ., πώς το Moq μιμείται το Repository Pattern στο Unit Test).
- Visualization Tools: Χρησιμοποιήστε εργαλεία όπως το Excalidraw ή το Miro για να σχεδιάζετε αρχιτεκτονικές και να τις συγκρίνετε με αυτές της ομάδας σας.

# 3. PERSONALIZED MODE

## Προτεινόμενα Μαθήματα (SerpAPI)

### Unit Testing

- [Top Unit Testing Courses Online - Updated \[November 2025\]](#)
- [Introduction to Unit Testing](#)
- [JavaScript Unit Testing - The Practical Guide](#)
- [Top JUnit Courses Online - Updated \[November 2025\]](#)
- [Unit Testing for Typescript & Node.js Developers with Jest](#)
- [Unit Testing: What It Is and How It's Done](#)
- [Best JUnit Courses & Certificates \[2025\]](#)
- [Complete Guide to Unit Testing in .NET Core \(NUnit & xUnit\)](#)
- [Beginner's Guide to Unit Testing](#)
- [Unit Testing in Java](#)
- [Complete Software Testing Course - Unit Testing | Full Crash ...](#)
- [JavaScript Unit Testing Tutorial for Beginners](#)
- [Java Unit Testing with JUnit - Tutorial - How to Create And ...](#)
- [Angular Unit Testing Crash Course | Unit Testing for ...](#)
- [C# Unit Testing - Full Course - Write Unit Tests in C# like a pro!](#)

### Docker

- [Top Docker Courses Online - Updated \[November 2025\]](#)
- [Top Free Docker Courses & Tutorials Online](#)
- [Docker for the Absolute Beginner - Hands On - DevOps](#)
- [Docker & Kubernetes: The Practical Guide \[2025 Edition\]](#)
- [Docker Course from Basics to Advanced \[Full course\]](#)
- [Best Docker Courses & Certificates \[2025\]](#)
- [Docker for Beginners with Hands-on labs](#)
- [Introduction to Docker](#)
- [Introduction to Containers w/ Docker, Kubernetes & ...](#)
- [What Is The Best Way to Learn Docker?](#)
- [Complete Docker Course - From BEGINNER to PRO! \(Learn ...\)](#)
- [Day 1/40 - Docker Tutorial For Beginners - Docker ...](#)
- [Learn Docker in 1 Hour | Full Docker Course for Beginners](#)
- [Learn Docker in 2 Hours - A Full Tutorial \(2025\)](#)
- [Docker Crash Course for Absolute Beginners \[NEW\]](#)

## SOLID Principles

- [Top SOLID Principles Courses Online](#)
- [SOLID Principles of Object Oriented Design and Architecture](#)
- [SOLID Principles: Introducing Software Architecture & Design](#)
- [Low Level System Design, Design Patterns & SOLID ...](#)
- [SOLID Principles - Free Software Engineering Tutorial](#)
- [ASP.NET Core - SOLID and Clean Architecture \(.NET 5 ...](#)
- [Structural Design Patterns in Modern C++](#)
- [Best Object Oriented Design Courses & Certificates \[2025\]](#)
- [Object Oriented Programming Courses Online](#)
- [Creational Design Patterns in Modern C++](#)
- [Learn SOLID Principles with CLEAN CODE Examples](#)
- [SOLID Principles: Do You Really Understand Them?](#)
- [Master Design Patterns & SOLID Principles in C# - Full OOP ...](#)
- [SOLID Design Principles | Complete Guide with Code Examples](#)
- [Uncle Bob's SOLID Principles Made Easy - In Python!](#)

## REST API

- [Top REST API Courses Online - Updated \[November 2025\]](#)
- [REST API Design, Development & Management](#)
- [Postman: The Complete Guide - REST API Testing](#)
- [Node.js Crash Course: Build a REST API in a Weekend](#)
- [Spring Boot REST APIs Ultimate Course](#)
- [Best RESTful API Courses & Certificates \[2025\]](#)
- [Building RESTful APIs Using Node.js and Express](#)
- [APIs](#)
- [REST APIs with Flask and Python in 2024](#)
- [Mastering REST APIs with FastAPI](#)
- [REST API Crash Course - Introduction + Full Python API Tutorial](#)
- [REST API Tutorials](#)

**ΣΑΣ ΕΥΧΑΡΙΣΤΩ  
ΠΟΛΥ!**