

نام دوره:

API Architecture Styles(Create Modern Application With REST,gRPC,GraphQL in .Net 8)

اهداف دوره:

در این دوره آموزشی جامع، ما به بررسی و تحلیل سه سبک محبوب و موثر معماری API در دنیای تکنولوژی امروز یعنی REST, gRPC و GraphQL خواهیم پرداخت. هدف اصلی این دوره، مسلح کردن توسعه‌دهندگان به دانش کافی برای ایجاد راحل‌های قابل اطمینان و کارآمد API است که پیشرفته‌ترین و مرتبط‌ترین تکنیک‌های موجود در .NET 8 را به کار می‌گیرد. شرکت‌کنندگان یاد خواهند گرفت که چگونه هر یک از این سبک‌ها را با درکی عمیق از مبانی، بهترین شیوه‌ها و تفاوت‌های کاربردی آن‌ها، به صورت عملی پیاده‌سازی کنند.

علاوه بر مفاهیم و استراتژی‌های تئوری، این دوره شامل موارد عملی فراوانی است که فرصت‌هایی برای تمرین و تقویت مهارت‌های توسعه و اجرای API در سناریوهای واقعی را فراهم می‌کند. تاکید ویژه‌ای بر ارزیابی امنیتی، نگهداری و مقیاس‌پذیری API‌ها خواهد بود تا شرکت‌کنندگان برای مواجهه با چالش‌های به هنگام در محیط‌های توسعه نرم افزاری مجهز شوند. در پایان این دوره، توسعه‌دهندگان نه تنها در ایجاد API‌های سازگار و کارآمد مهارت خواهند داشت، بلکه توانایی اتخاذ تصمیمات فنی مناسب برای نیازهای خاص پروژه خود را نیز کسب می‌کنند.

مخاطبین دوره:

تمامی افرادی علاقمند به یادگیری مباحث معماری و تولید برنامه‌های مقیاس‌پذیر می‌باشند، می‌توانند در این دوره شرکت نمایند. در بازار کار امروز دیگر دانستن کلی انتظارات کارفرما را برطرف نمیکند و افراد باید مهارت‌های زیادی یاد گیرند. ما در سماتک بر اساس تجربه و شناخت دقیق بازار کار دوره‌ای پروژه محور را طراحی کردیم.

پیشنیاز دوره:

گذراندن دوره C# پیشرفته

مدت دوره:

دستاوردها:

با شرکت در این دوره آموزشی، دانشجویان ابتدا با مقدمات و کلیات مربوط به API و اصول طراحی سرویس‌های وب آشنا خواهند شد. آنها مهارت‌های لازم برای ایجاد رابط‌های برنامه‌نویسی کاربردی و کارآمد با استفاده از سبک‌های معماری REST, gRPC و GraphQL را در محیط .NET 8 فرا خواهند گرفت. تمرکز عمده‌ی دوره بر تحلیل و مقایسه‌ی این روش‌ها با هدف درک عمیق‌تر مزایا و معایب هر کدام از آنها در کنار آشنایی با موقعیت‌های کاربردی مناسب برای هر سبک است.

در طول این دوره، دانشجویان به صورت عملی با پیاده‌سازی API های واقعی از طریق تمرین‌ها و پروژه‌های کاربردی، تجربه‌های مهمی کسب خواهند کرد. این تمرین‌ها به آن‌ها کمک می‌کند تا مهارت‌های خود در زمینه تست، اعتبارسنجی، امنیت، و مستندسازی API ها را تقویت نمایند. مهمتر از همه، این دوره به آن‌ها امکان می‌دهد که درک کنند چگونه API ها را به صورتی که با سیستم‌های کلاینت مختلف سازگاری داشته باشند، طراحی و توسعه دهند. این امر برای دانشجویان فرصتی برای توسعه مهارت‌های مربوط به بهینه‌سازی و عیب‌یابی در محیط‌های پیچیده و چندلایه‌ی نرم‌افزاری را فراهم می‌آورد.

پس از پایان این دوره، دانشجویان قدرت تحلیل و انتخاب استراتژی‌های مناسب برای توسعه API ها متناسب با نیازهای خاص هر پروژه را خواهند داشت. توانایی آنها در پیاده‌سازی راحل‌های مقیاس‌پذیر و پایدار که می‌توانند بار کاری بالا را مدیریت کرده و ارتباطات بین سرویس‌ها را بهینه‌سازی کنند، پرورش می‌یابد. در نهایت، دانش‌آموختگان این دوره با دانشی جامع و آمادگی کامل برای مواجهه با چالش‌های مربوط به معماری API در دنیای واقعی نرم‌افزار، وارد بازار کار خواهند شد.

سرفصل مطالب آموزشی:

API Architecture Styles

1. Introduction to APIs and Web Services

- a. Understanding API fundamentals
- b. Overview of Web Services
- c. The role of HTTP in Web APIs

2. RESTful API Design

- a. REST principles and Best Practices
- b. Designing RESTful services
- c. Implementing CRUD operations in .NET 8

- d. Error handling and status codes
- 3. Building and Consuming gRPC Services**
 - a. Introduction to gRPC and Protocol Buffers
 - b. Defining gRPC services and messages
 - c. Implementing gRPC in .NET 8
 - d. Comparison with REST: Performance and Usability
- 4. Implementing GraphQL APIs**
 - a. Understanding GraphQL queries, mutations, and subscriptions
 - b. Designing and constructing a GraphQL schema
 - c. Optimizing data fetching with GraphQL
 - d. Security considerations in GraphQL
- 5. API Security and Best Practices**
 - a. Authentication and Authorization mechanisms
 - b. Securing API endpoints
 - c. API Keys, OAuth, and JWT tokens
 - d. CORS policies and best practices
- 6. Testing and Documentation**
 - a. Unit Testing of API Endpoints
 - b. Integration Testing Strategies
 - c. Automated Documentation Tools (Swagger/OpenAPI)
- 7. API Performance Tuning and Optimization**
 - a. Caching strategies and content compression
 - b. Rate Limiting and Throttling
 - c. Profiling and Load Testing
- 8. API Versioning and Evolution**
 - a. Strategies for versioning APIs
 - b. Handling breaking changes
 - c. Deprecating API endpoints

Entity Framework

1. Introduction to Entity Framework

- a. Definition and purpose of Entity Framework.
- b. Role of EF in database interactions.
- c. Advantages of using EF in C# applications.
- d. EF Versions

2. Getting Started with EF in C#

- a. Setting up an EF project in Visual Studio.
- b. Configuring the database connection.
- c. Creating a simple data model using EF annotations.

3. Code-First Approach

- a. Explanation of the Code-First development approach.
- b. Creating entities, relationships, and configuring the database using code.
- c. Migrations and updating the database schema.
- d. Code-First vs Database-First

4. CRUD Operations with EF

- a. Performing Create, Read, Update, and Delete operations using EF.
- b. Writing LINQ queries for data retrieval.
- c. Saving changes to the database.

5. Querying and Filtering Data

- a. Utilizing LINQ to Entities for querying data.
- b. Filtering and projecting data in EF queries.
- c. Understanding lazy loading and eager loading.

6. Optimizing Performance with EF

- a. Techniques for optimizing EF queries.
- b. Using caching and reducing database round-trips.
- c. Handling large datasets efficiently.

7. Concurrency and Transactions

- a. Managing concurrent data access with EF.
- b. Understanding optimistic and pessimistic concurrency.
- c. Implementing transactions in EF.