

حُكُومَةُ الشَّارِقَةِ  
دائرة التنمية الاقتصادية



How to develop a product?

# Web & Mobile Platforms

<https://suhayeb.com/services/coaching/product-development>

Updated Course Content

# *Product Development*

Methodologies, Architecture, Backlog

# 1 Product Development

The process of bringing a new product or improving an existing one to meet customer needs and market demands



## Solution to Problem

# Evolution

Progressive evolution of global information that users can access computer connected to net, differs in **Architecture, Technologies, User Interaction, Data Ownership** and **Monetization**

## Web 4.0?

### Web 1.0 read-only

Static websites providing information without much interactivity. Users were mostly consumers.

### Web 2.0 social

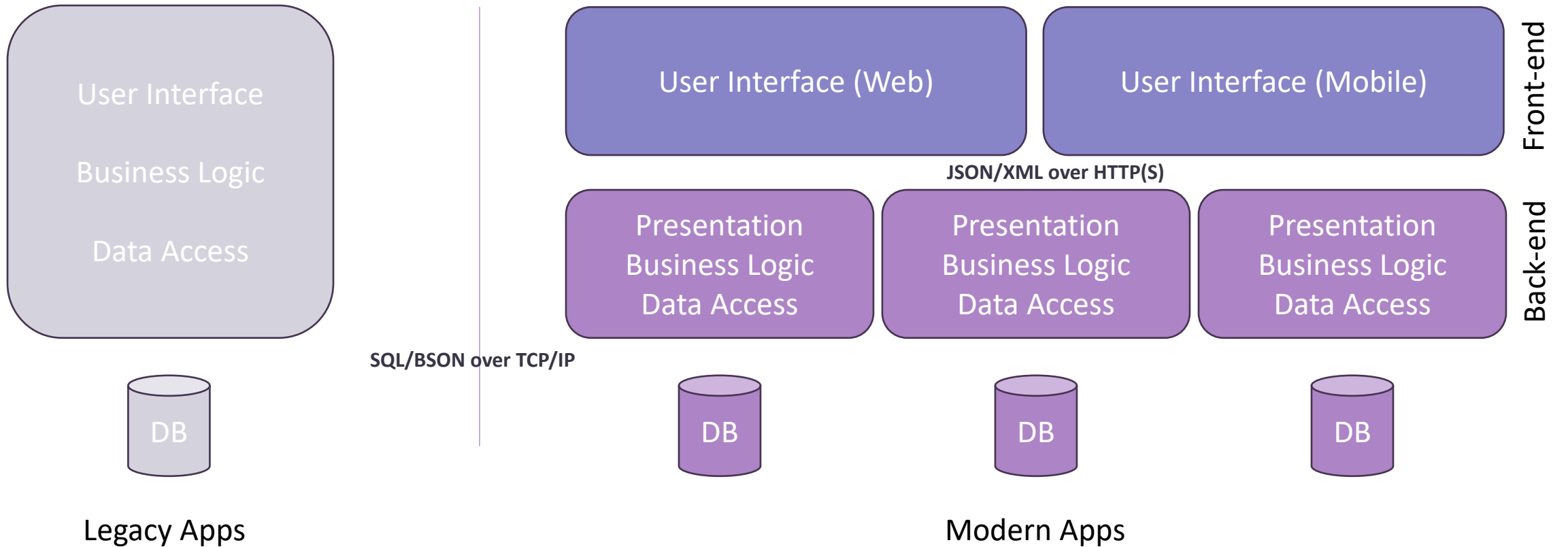
Interactive and dynamic web with user-generated content, social networking, and collaboration



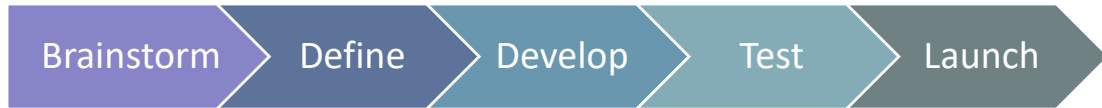
### Web 3.0 semantic

Decentralized and intelligent, focusing on data ownership, blockchain technology.

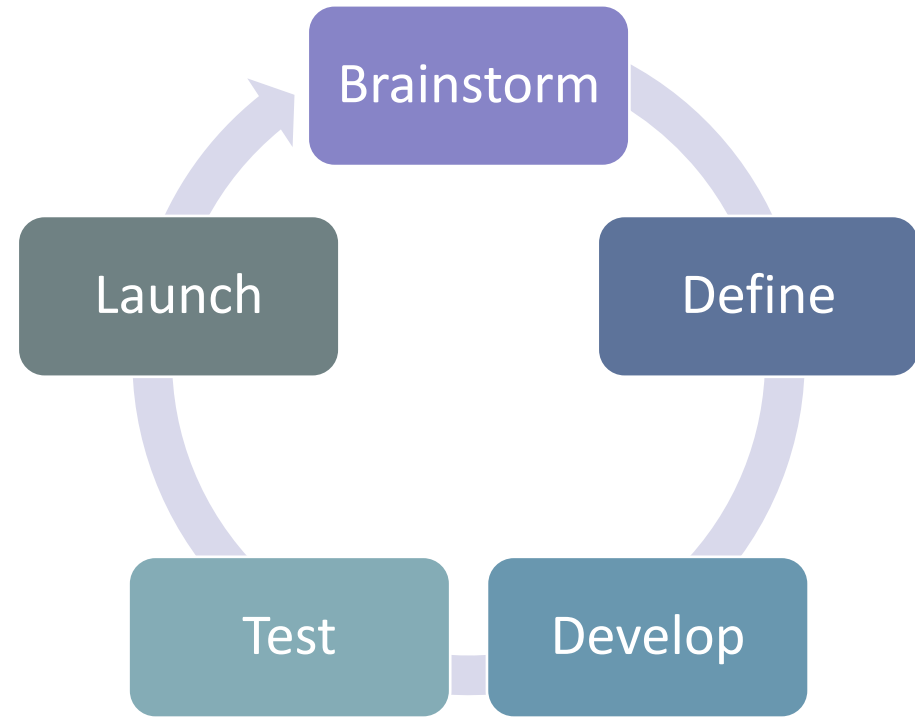
# Web 2.0 Architectures



# Methodologies



Waterfall

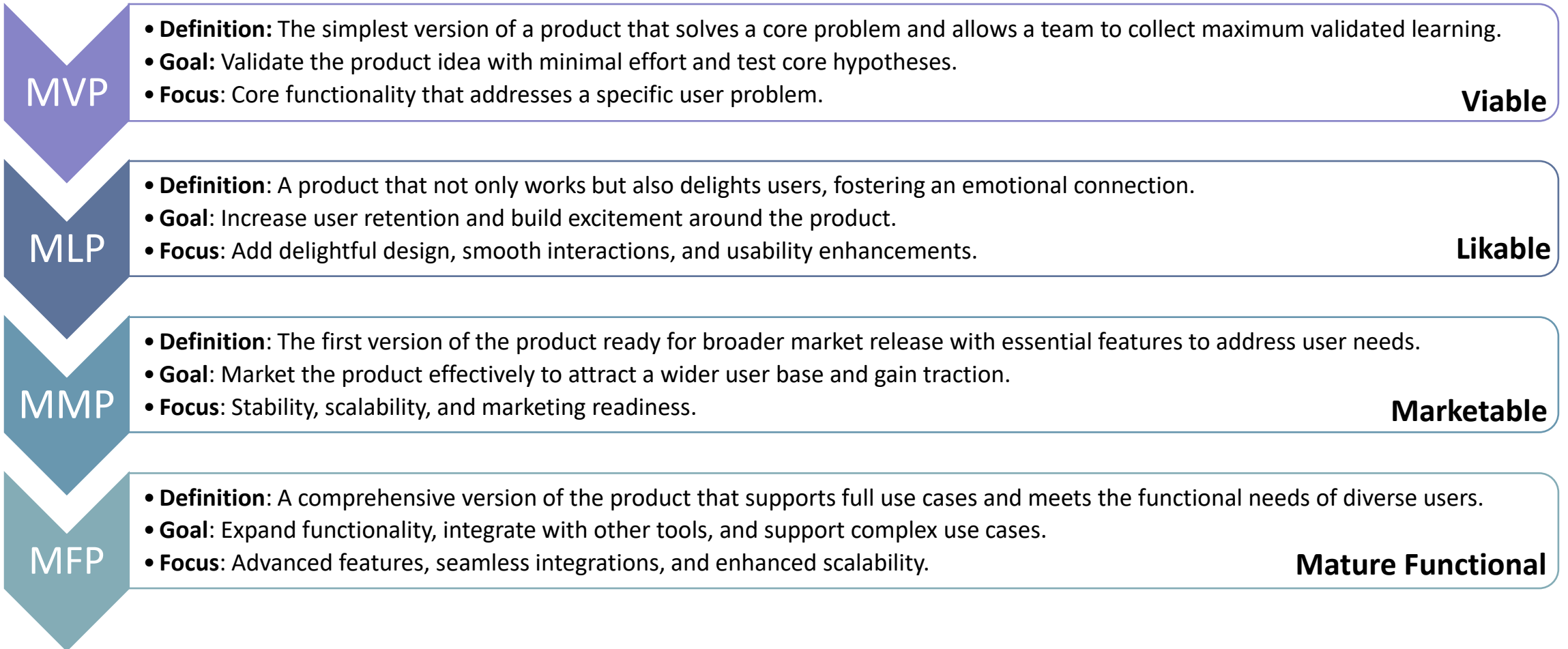


Agile

# Methodologies

	Waterfall (Traditional)	Agile
<b>Process Flow</b>	Sequential / Linear	Iterative (Sprints)
<b>Flexibility</b>	Fixed (Scope, budget)	Evolves based on feedback
<b>Collaboration</b>	Silos, Emphasis on Documentation	High among cross-functional, continuous
<b>Delivery Timeline</b>	Project timeline	Typically, every 2 weeks
<b>Risk</b>	High due to late test and user feedback	Low due to continuous testing
<b>Customer Involvement</b>	Limited until the end	Continuous involvement
<b>Known Frameworks</b>	PMBOK	SCRUM, SAFe

# Deliverables





# Prioritization

- Understand the Job Hierarchy:
  - Core Jobs (functional needs, Primary and Secondary)
  - Emotional Jobs (how users feel)
  - Social Jobs (how users are perceived by others)
- Use Questions to Prioritize Features:
  - What is the primary job this feature helps accomplish?
  - Does this feature solve a major pain point or enhance an existing solution?
  - Does this feature align with the desired outcome of this development phase (MVP, MLP, etc.)?

People **Hire**

**Product** to ..

**Jobs To Be Done**

# *AI Use in Product Development*

Topics

## 2 AI Use


- Brainstorming
- Generating code (components, functions)
- Generating data (dummy, test, etc..)
- Code auto complete
- Debugging
- Documentation / explanation
- Translation among languages
- Testing

# AI Tools

- **Github Co-Pilot**
- ChatGPT
- Checkout:
  - Builder.io: Figma to code
  - Codeium (Free Github CoPilot)
  - Tabnine

# GitHub CoPilot



- **GitHub Co-Pilot (Free Tier, Now Available)**
- Main features:
  - Launch on Visual Code 
  - Right click anywhere, Ask GPT in your own language, just be clear
  - Use @workspace for AI to understand your whole project and generate code according.

# *Code Management*

Topics

# Code Management



**Uses: download & Install** GIT ([git-scm.com](https://git-scm.com))

Platforms:

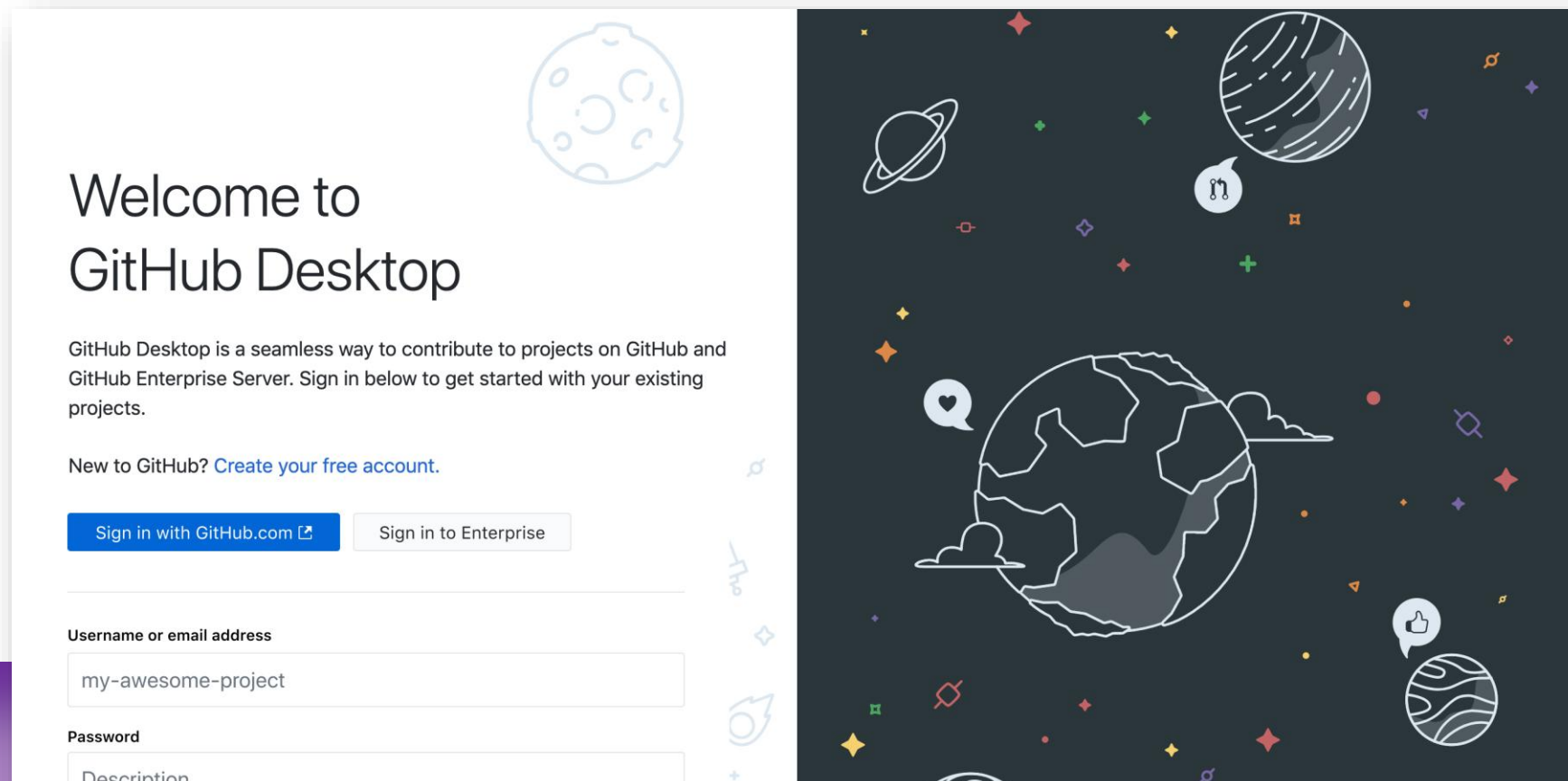
- Github (Community & Code files Hosting) well known for their integrations
- BitBucket.

Signup today, join coders community

# Code Management



**download & Install Github Desktop** ([desktop.github.com/download/](https://desktop.github.com/download/))





# *Project*

Digital Nomad License

# 4 The Project

Allow applicants to apply for freelance licenses, enable authorities to validate them, and showcase an approved public list for businesses to connect with nomads



## Digital Nomad License

# Components

During this course, we will use current technologies to build the 1<sup>st</sup> version of the product



## Web API Framework

A powerful, scalable, and secure backend framework for building robust web APIs.



## JavaScript Framework

for building dynamic and interactive user interfaces.



## CSS Framework

responsive design and prebuilt components for faster development



## Flutter

## Cross-Platform Framework

A versatile UI toolkit for building natively compiled apps for mobile, web, and desktop.

# *Front-End*

Topics

# Front-end Frameworks

Simplifies UI development with reusable components, efficient DOM handling, and responsive designs. They enable faster development, consistent styling, and seamless scalability for modern web apps.



## CSS Frameworks

### Tailwind

Custom UIs; utility-first, highly customizable, lightweight.

### Bootstrap

Prototypes, admin panels; pre-built, responsive components.

### Bulma

Blogs, small apps; modern, clean, and Flexbox-based design.

## JS Frameworks

### React (Facebook)

SPAs and UIs; flexible, component-based, vast ecosystem.

### Angular (Google)

Enterprise apps; robust tools, full-featured framework.

### VueJS

Lightweight apps; easy integration, simple, and flexible.

# Front-end Environment

- **Download & Install:** Node.js ([nodejs.org](https://nodejs.org))
- **Download & Install :** Visual Studio Code ([code.visualstudio.com](https://code.visualstudio.com))
- **Execute:** (Requires elevated permission on command line and powershell)
- `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`
- `npm install -g pnpm`
- **Download & Install :** Screaming Frog Spider 😊

# 3 Cross-platform: Vue

## Should know

- A prototype for user journey with basics of Vue/Nuxt
- Reactive data binding (using ref and reactive).
  - Declarative rendering with templates.
  - Directives (v-for, v-if, v-bind :, v-on @, v-html etc.).
  - Event handling @submit modifier .prevent, @click, @mouseenter.
  - Two-way binding with v-model.
  - Component Architecture
  - Lifecycle Hooks.
  - Builtin: watch, computed
- State management
  - Routing
  - Props

Course Link



# 5 Web Searchability

It is important that SEO practices are considered when developing product

- **Core Elements:** Title (60), Robots.txt, Sitemap.xml
- **Meta data:** Description (160), Keywords, Open Graphs, Twitter Cards
- **HTML Structure,** H1, H2, H3 etc..
- **Image Alternative Text**
- **Internal Links,** the more the marrier
- **Schema Structure**
- **Errors**




PageSpeed Insights





# Hosting


Publishing Front end application, can be done automatically if hosted on Github



**netlify**

**Netlify.com**

Best for static sites and JAM stack apps (Angular, React, Vue)



**vercel.app**

Optimized for Next.js apps (React)

# *Backend*

Topics

# Environment

- **Download:** Visual Studio Community ([visualstudio.microsoft.com](https://visualstudio.microsoft.com))
- **-- Install:** *ASP.NET and web Development* workload

# Database Platforms

	MySQL	PostgreSQL	MongoDB
Data Model	Tables with fixed schemas	Tables with support object	Documents
Performance	Optimized for read-heavy workloads	Optimized for complex queries and concurrent writes	Fast for unstructured or semi-structured data
Query Language	SQL	SQL Advanced	MQL (~JSON)
Best for	Beginners and standard relational data needs	Developers needing advanced data modeling and queries	Projects requiring schema flexibility and scalability

**MySQL**  
Relational Database Management System

**PostgreSQL**  
Object-Relational Database Management System

**MongoDB**  
NoSQL Document Database

# Database

## Intro

### What is Database:

A database is an organized collection of data stored and managed electronically, enabling efficient retrieval, manipulation, and storage of information for various applications.

### Core concept - Normalization:

Normalization is a process in database design aimed at organizing tables to reduce redundancy and ensure data integrity by dividing data into related tables and defining relationships.

#### MySQL

Relational Database Management System

#### PostgreSQL

Object-Relational Database Management System

#### MongoDB

NoSQL Document Database

# Back-end Frameworks

## Node.js (Express.js)

Best used on I/O heavy apps, chats, streaming etc..

## Django (REST, Python)

Best used for scientific, e-commerce apps.

## Spring Boot (Java)

Best used for large-scale e-commerce, enterprise-level applications

## .NET (Express.js)

Best used for Enterprises and large organizations

	Express.js	Django	SpringBoot	.NET
Language	JavaScript	Python	Java	C#
Performance	High (I/O)	Moderate	High	High
Learning Curve	Easy	Moderate	Steep	Moderate
Tools	Limited	Limited	Strong (Eclipse)	Best (Visual Studio)
Security	Requires manual works	Strong (Builtin)	Strong	Strong (OOTB)
Scalability	Excellent	Good	Excellent	Excellent



# 3 Cross-platform: .NET API

## Should know

- Setup: Create projects and configure middleware.
- Routing: Use attribute routing and versioning.
- Controllers: Handle requests and return responses.
- Models: Define and validate data structures.
- Database: Integrate Entity Framework Core for CRUD.
- Dependency Injection: Manage services effectively.
- Authentication: Implement JWT and role-based access.
- Error Handling: Handle exceptions and log errors.
- Async: Use async/await for scalability.
- Docs & Testing: Generate Swagger docs and test APIs.

Course  
Link



# Hosting

Publishing Front end application, can be done automatically if hosted on Github



**Azure.com**

Comprehensive Cloud  
platform for hosting .NET  
Apps



**docker.com**

Hosting on your computer,  
launching apps in isolated  
containers (mini server)



# *Cross-Platforms*

Topics

# Cross-platform

## Development platform

	Flutter	React Native	MAUI
Language	Dart	JavaScript	C#
Use cases	Mobile-first applications with complex UIs. MVPs and startups needing fast development cycles.	Cross-platform mobile apps needing high customizability, especially for web apps using React	Enterprise applications, especially those in the Microsoft ecosystem.
Learning Curve	Moderate	Easy	Moderate

### Flutter (Google)

Single codebase for iOS, Android, web, and desktop (Windows, macOS, Linux).

### React Native (Facebook)

Single codebase for iOS and Android.

### .NET MAUI (Microsoft)

Single project structure for iOS, Android, Windows, and macOS.

# Flutter Environment

- **Download & Install :** Visual Studio Code ([code.visualstudio.com](https://code.visualstudio.com))
- **Download & Install :** Android Studio  
(<https://developer.android.com/studio/install#windows>)
- **Download & Install:** Flutter SDK (<https://docs.flutter.dev/get-started/install/windows/mobile>) and follow instruction

# 3 Cross-platform: Flutter

## Should know

- Setup: Install Flutter SDK and create a new project.
- Widgets: Use Stateless and Stateful widgets.
- Layouts: Build UIs with Column, Row, Stack, and Flex.
- State Management: Manage state with Provider, Riverpod, or Bloc.
- Navigation: Implement routing and multi-page navigation.
- Forms: Build forms with validation and user input handling.
- Networking: Fetch data with http or dio.
- Async: Use Future and Stream for asynchronous tasks.
- Packages: Add functionality using pub.dev packages.
- Testing: Write unit, widget, and integration tests.

Course  
Link

