

Full Stack (400 Academic hours)

HTML5

- Elements and Attributes
- Text (Heading, Paragraph)
- Styles
- Divs
- Images and Media
- Forms
- Tables
- Working with Embed code / iFrames
- Links
- Lists
- Responsive
- HTML API (Drag/Drop, Geolocation, Storage)
- Design systems: Bootstrap / Material

CSS3

- Selectors, Classes
- Responsive (viewport, grid, queries)
- Working with Colours (solid, gradient)
- Text styling
- Flex
- Positions
- Variables
- Containers
- Effects
- Animations

TypeScript

- Built-in types
- Variables
- Arrays
- Tuples
- Enums
- Functions
- Objects
- Type Aliases
- Union / Intersection / Literal / Nullable Types
- Optional Chaining
- Classes
- Getters and Setters
- Methods
- OOP (Inheritance / Polymorphism)
- Interfaces & Types
- Generic
- **Working with NPM - Axios, Moment, Geodist**

React v18

- Setup
- Components
- React Ecosystem
- Fragments
- Events
- ListGroups
- Managing state
- Props (passing data and functions)
- React Dev Tools
- Style (Vanilla / Modules / JS)
- UI Libraries (Bootstrap / Chakra UI)
- Forms
- Working with API (Axios)
- Navigation
- Working with Firebase (Authentication, Storage, Hosting, Firestore)
- Hooks (useState, useEffect, useRef, useContext, useCallback)
- Global state management (Zustand)
- React Query

React Native

- Setup (Environment & Emulators)
- Working with EXPO
- React native built-in components
- Navigation (v6)
- Design UI Libs: React native paper / elements
- Advance features: Camera and Gallery / Geolocation & MapView / Sharing / Contacts / Calendar / Permissions / Dates / Push Notifications
- AsyncStorage
- Media
- Lottie
- SQLite
- Publish to AppStore & Google Play

NEW! Server Node with TypeScript

- Core modules (fs, http, path, etc.) and their TypeScript typings
- Creating a simple HTTP server with TypeScript
- Setting up Express with TypeScript
- Typing Express middleware, requests, and responses
- Building REST APIs with Express and TypeScript
- Error handling in Node.js and TypeScript
- Debugging TypeScript with source maps
- Integrating SQL databases (e.g., PostgreSQL) with Sequelize or TypeORM
- Integrating NoSQL databases (e.g., MongoDB) with Mongoose
- Typing database models and queries
- Implementing JWT-based authentication
- Role-based access control (RBAC) in Node.js applications
- Unit testing with Jest
- Integration testing APIs
- WebSockets with Socket.io
- Real-time features in REST APIs
- Decorators and Metadata Reflection

- Working with Postman & Swagger

NEW! AI and Machine Learning

- Cover the fundamental concepts, types of algorithms, and how they can be applied to solve real-world problems.
- TensorFlow.js - adding machine learning models to web applications.
- Integrating AI Services with Full-Stack Applications
- integrate third-party AI APIs (e.g., Google Cloud AI, IBM Watson, Azure Cognitive Services)
- Implementing chatbots using AI APIs or libraries for customer support or information retrieval.
- Building recommendation systems for e-commerce or content websites using ML algorithms.
- Using AI for facial recognition, object detection, or automated video summarization in web applications.

Databases

Firestore Firestore

- Overview and Architecture
- Firestore vs. Realtime Database
- Documents and Collections
- Data Types
- CRUD Operations
- Simple and Compound Queries
- Indexing and Composite Indexes
- Relationships: One-to-One, One-to-Many, Many-to-Many
- Security Rules Basics
- Real-time Data **NEW! PostgreSQL**
- Overview and Features
- PostgreSQL vs. Other Databases
- SQL Basics and Advanced Queries
- Joins, Subqueries, and Window Functions

- Primitive Types
- Arrays, JSONB, and Other Advanced Types
- Index Types
- ACID Properties
- Stored Procedures and Triggers
- Streaming Replication **MongoDB**
- Overview and Core Concepts
- Document-Oriented Model
- CRUD Operations
- Schema Design Principles
- Embedded vs. Referenced Relationships
- Index Types and Creation
- Performance Tuning
- Pipeline Stages
- Sharding Strategies
- Authentication and Authorization

NEW! DevOps Fundamentals

- Understanding DevOps: Principles and Practices
- Continuous Integration (CI), Continuous Delivery (CD), and Continuous
- Jenkins: Automating Builds and Deployments
- Jenkins UI and Basic Concepts
- Building a Simple CI/CD Pipeline
- Integrating Automated Tests
- Setting Up Build Triggers
- Distributed Builds with Jenkins Agents
- Understanding Containers and Docker
- Creating Dockerfiles
- Building, Tagging, and Pushing Images
- Defining Multi-Container Applications with Docker Compose
- Container Networking and Volume Management
- Overview of AWS Developer Tools (CodeCommit, CodeBuild, CodeDeploy,

CodePipeline)

- Integrating Jenkins, Docker, and AWS

מבין לקוחותינו:

J.P.Morgan

