

# Revised Java Full Stack + GenAI + AgentAI Curriculum (2025 Edition)

## Month 1: AI-Enhanced Frontend Development

Week	Topic	Updated Modules
1	<b>HTML, CSS, and GenAI-assisted Design</b>	Introduction to AI-enhanced Web Design, Using AI tools like Galileo AI and Figma AI for UI wireframes, Building responsive pages with Tailwind CSS
2	<b>JavaScript Fundamentals + AI Code Assistants</b>	Core JS concepts, AI Pair Programming using GitHub Copilot, Debugging with CursorAI
3	<b>Modern UI with React + AI Prompts</b>	React Basics, Hooks, AI-assisted component generation (v0, Bolt.new), Building reusable UI with GenAI
4	<b>No-Code UI Builders</b>	Bubble, Framer AI, Webflow AI; Integrating with React codebase

---

## Month 2: Backend Development with AI Support

Week	Topic	Updated Modules
1	<b>Java + Spring Boot (AI-assisted)</b>	Modern Java Syntax, Copilot for backend boilerplate, REST API structure using AI suggestions
2	<b>Database Integration</b>	MySQL, MongoDB; Query generation using ChatGPT; AI-based schema design tools
3	<b>API Development + Agentic AI Integration</b>	REST APIs, API testing via Postman AI, Integrating LLM endpoints using OpenAI API
4	<b>No-Code Backend Platforms</b>	Xano, Backendless, Supabase AI features

---

## Month 3: Full Stack + GenAI Integration

Week	Topic	Updated Modules
------	-------	-----------------

1	<b>Connecting Frontend &amp; Backend</b>	API consumption, Authentication, AI-assisted debugging
2	<b>Generative AI for Developers</b>	Prompt Engineering, ChatGPT API, Hugging Face tools
3	<b>LLM-based App Development</b>	Build custom Chatbot with LangChain.js & React
4	<b>AI Tools for Productivity</b>	Cursor, Replit AI, Codeium, Copilot Workflows

---

#### Month 4: Agentic AI + Automation

Week	Topic	Updated Modules
1	<b>Understanding Agentic AI</b>	Concept of AI Agents, Multi-agent frameworks (CrewAI, Autogen)
2	<b>No-Code AI Agents</b>	Use of AgentDK, Chatflow AI, Zapier + GPT integrations
3	<b>Building Intelligent Assistants</b>	Task automation with CrewAI, AutoGPT
4	<b>Mini Project</b>	Build a React + Spring Boot App with AI-powered recommendations

---

#### Month 5: Cloud + DevOps + GenAI Deployment

Week	Topic	Updated Modules
1	<b>Cloud Fundamentals</b>	AWS Amplify, Vercel, Render, Firebase
2	<b>AI Deployment</b>	Deploying LLM-based apps
3	<b>CI/CD Pipelines (AI-Automated)</b>	GitHub Actions with Copilot
4	<b>Mini Project</b>	Build a full-stack GenAI App with deployment

---

#### Month 6: Capstone + Industry Readiness

Week	Topic	Updated Modules
1	Capstone Project Planning	Use AI tools for idea validation & architecture
2	Implementation Phase	GenAI + AgentAI integrations
3	Testing & Evaluation	AI tools for unit tests, Postman AI
4	Final Presentation	Deploy & demo AI-driven product

---

## Month 1: AI-Enhanced Frontend Development

**Objective:** Build strong foundations in frontend technologies (HTML, CSS, JS, React) while integrating Generative AI and No-Code tools for faster prototyping and intelligent UI building.

---

### Week 1 – HTML, CSS, and AI-Assisted UI Design

Lecture	Topic	Description
1.1.1	Introduction to Full Stack + GenAI Design Tools	Overview of Full Stack roadmap; Introduction to Galileo AI, Figma AI, and Uizard for UI mockups; Hands-on: Auto-generating UI using prompts
1.1.2	HTML Basics + AI Pairing	HTML Structure, Semantic tags; Use of ChatGPT/Cursor to generate HTML boilerplate; Project: Portfolio Page
1.1.3	Advanced HTML5 + Media Tags	Forms, Audio/Video tags, Accessibility; Using AI tools to generate accessible markup

1.1.4	<b>CSS Foundations</b>	Inline, Internal, External CSS; Styling via ChatGPT-generated snippets
1.1.5	<b>CSS3 + Animations</b>	Transitions, Transformations, Gradients; AI-assisted creation of animations via Uizard or Figma plugins
1.1.6	<b>Responsive Design with Flex/Grid</b>	Flexbox, Grid, Media Queries; Generate layouts using Galileo AI
1.1.7	<b>Mini Project 1 – AI-Powered Portfolio Page</b>	Use Galileo AI + HTML/CSS + Tailwind to build an auto-generated portfolio page

#### **In-Class Activities:**

- AI prompt-based UI creation using Uizard
- Responsive layout exercise using Flexbox Playground

#### **Post-Class Assignments:**

- Create a 3-section homepage using AI tools for wireframe generation
- Publish your design using Vercel

---

## **Week 2 – JavaScript Fundamentals + AI Pair Programming**

<b>Lecture</b>	<b>Topic</b>	<b>Description</b>
1.2.1	<b>Intro to JavaScript + AI Debugging</b>	Syntax, Variables, Data Types; AI Copilot-assisted debugging
1.2.2	<b>Functions, Arrays, Objects</b>	Real-time code suggestions using Cursor; Hands-

on: Employee data script

1.2.3	<b>Loops &amp; Conditional Logic</b>	Use AI to convert pseudocode to JS loops
1.2.4	<b>DOM Manipulation</b>	Selecting, modifying, and creating elements dynamically; AI tool: Copilot
1.2.5	<b>Events &amp; Forms</b>	Form validation using AI-generated snippets
1.2.6	<b>Error Handling &amp; Best Practices</b>	Identify and fix JS bugs using AI explainers
1.2.7	<b>Mini Project 2 – AI-enhanced Quiz App</b>	Use Cursor or Replit AI to build a dynamic quiz app

**In-Class:**

- Use ChatGPT to explain JS logic
- Build DOM manipulation snippets with Cursor

**Post-Class:**

- AI-assisted code reviews (students analyze Copilot-generated suggestions)

---

## Week 3 – React Fundamentals + GenAI Components

Lecture	Topic	Description
1.3.1	<b>React Introduction</b>	Why React? Component-driven UI; AI-assisted setup using Replit AI

1.3.2	<b>JSX + Props + State</b>	Build simple components; Use v0.dev for AI component scaffolding
1.3.3	<b>Events + Lists + Keys</b>	Create Todo List using AI-suggested logic
1.3.4	<b>React Hooks</b>	<code>useState</code> , <code>useEffect</code> ; Auto-generate hook logic with ChatGPT
1.3.5	<b>AI-Generated Components</b>	Generate React components via v0.dev & Bolt.new
1.3.6	<b>Routing and Navigation</b>	React Router setup via AI-generated boilerplate
1.3.7	<b>Mini Project 3 – AI Blog UI</b>	Use AI to generate a blog homepage, integrate sample posts

#### In-Class:

- Explore Cursor's React template feature
- Use ChatGPT for error debugging

#### Post-Class:

- Build a responsive portfolio UI using v0.dev components

---

## Week 4 – No-Code UI Builders + Integration

Lecture	Topic	Description
1.4.1	<b>Introduction to No-Code Platforms</b>	Overview: Bubble, Framer, Webflow, and Glide

1.4.2	<b>Using Bubble AI</b>	Auto-generate UI and workflows using prompts
1.4.3	<b>Webflow AI for Responsive Design</b>	Build portfolio homepage via prompt
1.4.4	<b>Figma AI Prototyping</b>	Auto-generate variants and layouts using AI
1.4.5	<b>Integrating No-Code Apps with React</b>	Export Bubble/Webflow components into React projects
1.4.6	<b>AI Tools for UX Writing and Accessibility</b>	Use ChatGPT and GrammarlyGO for content generation
1.4.7	<b>Mini Project 4 – No-Code + React Integration</b>	Create a landing page in Webflow AI and integrate it in React

#### **In-Class:**

- Design + deploy a UI in Bubble AI
- Export Webflow design and link to React frontend

#### **Post-Class:**

- Create a mini portfolio using AI tools only
- Reflection: Compare manual vs AI-generated designs

---

## **End of Month Project: “AI-Powered Personal Portfolio”**

#### **Goal:**

Build a multi-section portfolio website that uses:

- HTML, CSS, JS, React
- Figma AI or Galileo AI for design
- Webflow/Bubble AI for layout generation

---

## Month 2: Backend Development with AI Support

### Objective:

Introduce Java and Spring Boot for backend development, leveraging AI-assisted coding and no-code backend platforms. Students will learn API creation, database integration, and AI-supported backend automation.

---

### Week 1 – Java Foundations and AI-Assisted Coding

Lecture	Topic	Description
2.1.1	Introduction to Backend Development	Understanding the backend architecture, API-driven design, client-server model, and AI tools used in backend development
2.1.2	Java Basics	Syntax, Data Types, Control Statements; Using ChatGPT or Copilot for code explanations and boilerplate generation
2.1.3	Object-Oriented Programming Concepts	Classes, Objects, Inheritance, Polymorphism; AI-assisted code examples and UML diagram generation
2.1.4	Collections and Generics	Working with Lists, Sets, Maps; Use Copilot for generic-based collection handling
2.1.5	Exception Handling	Try-catch-finally, custom exceptions; Using AI to suggest error messages and resolution
2.1.6	Java Streams and Lambda Expressions	Functional programming concepts; Generate stream operations using AI prompts

2.1.7	Mini Project 1 – AI-Paired Java Utility Builder	Build a CLI-based tool with Copilot; for example, a Student Record Manager using Lists and Maps
-------	---	---

#### In-Class Activities:

- Generate class structures using ChatGPT
- Debug Java programs with AI-assisted explainers

#### Post-Class Assignments:

- Write a Java class using AI for method recommendations
- Explore functional programming use cases using Copilot suggestions

---

## Week 2 – Spring Boot Fundamentals

Lecture	Topic	Description
2.2.1	Introduction to Spring Boot	Understanding dependency injection, Spring Core concepts, and project setup via Spring Initializr
2.2.2	Creating RESTful APIs	Using annotations like <code>@RestController</code> , <code>@GetMapping</code> , <code>@PostMapping</code> ; Generate endpoint code via AI
2.2.3	Dependency Injection and Beans	Understand <code>@Autowired</code> , scopes, and lifecycle; Use AI to visualize flow
2.2.4	Service Layer and Repository	Implementing business logic; Generating repository interfaces with Copilot
2.2.5	Handling HTTP Requests	Use AI tools to generate controller logic
2.2.6	Testing APIs	Using Postman and Postman AI for request validation and test case generation

2.2.7	Mini Project 2 – Employee CRUD API	Build a complete CRUD API for Employee data using Spring Boot and AI-suggested repository patterns
-------	------------------------------------	--

#### **In-Class:**

- Generate boilerplate controller code via Cursor AI
- Test API using Postman AI for validation

#### **Post-Class:**

- Write unit tests with AI-generated templates
- Deploy API locally and share endpoint collection

---

### **Week 3 – Database Integration and AI Query Builders**

Lecture	Topic	Description
2.3.1	Introduction to Databases	SQL vs NoSQL; choosing the right database for AI-based systems
2.3.2	MySQL Integration with Spring Boot	JDBC configuration, repository pattern, writing queries using AI
2.3.3	MongoDB and NoSQL Integration	Using MongoRepository; AI to assist with query operators
2.3.4	AI-Assisted Query Building	Use ChatGPT or Gemini to auto-generate CRUD queries
2.3.5	ORM with JPA and Hibernate	Introduction to JPA annotations and entity relationships
2.3.6	Database Design Automation	Tools like dbdiagram.io and ChatGPT for schema generation

2.3.7	Mini Project 3 – AI Inventory Management Backend	Build CRUD APIs for inventory items, integrating MySQL and MongoDB with AI-assisted schema design
-------	--	---

#### **In-Class:**

- Use ChatGPT to generate SQL queries from plain text
- Visualize schema with AI tools

#### **Post-Class:**

- Write entity classes and repositories using AI suggestions
- Share generated ER diagrams

---

### **Week 4 – APIs, AI Endpoints, and No-Code Backends**

Lecture	Topic	Description
2.4.1	REST API Deep Dive	Methods, Headers, Status Codes, and JSON Response Handling
2.4.2	API Documentation	Use Swagger UI and AI tools for automated documentation
2.4.3	Integrating AI APIs	Using OpenAI API and Hugging Face Inference API for simple GenAI use cases
2.4.4	Building Intelligent Endpoints	Create routes that use AI to generate dynamic responses
2.4.5	No-Code Backends Overview	Introduction to Backendless, Xano, Supabase AI
2.4.6	Building APIs using No-Code Tools	Use Xano to design, publish, and test APIs

2.4.7	Mini Project 4 – AI-Driven Feedback API	Build a feedback collection API that summarizes responses using OpenAI API
-------	---	--

#### **In-Class:**

- Integrate a simple AI summarization API
- Create and deploy APIs using Xano

#### **Post-Class:**

- Compare manual backend development vs no-code API generation
- Document AI API integration process

---

### **End of Month Project: “AI-Assisted Employee Management System”**

#### **Goal:**

Develop a backend API system for managing employees with AI support for data summaries and documentation.

#### **Key Features:**

- Built using Java, Spring Boot
- Database integration with MySQL or MongoDB
- AI-generated API documentation (Swagger + AI)
- Summarization endpoint using OpenAI API

#### **Deliverables:**

- Hosted API endpoints
- Swagger documentation

- GitHub repository
- Postman AI test suite

---

## Month 3: Full Stack + GenAI Integration

### Objective:

By the end of this month, learners will understand how to connect frontend and backend layers, integrate AI APIs, build LLM-powered applications, and use AI tools for improving productivity, debugging, and feature development.

---

### Week 1 – Connecting Frontend and Backend

Lecture	Topic	Description
3.1.1	Introduction to Full Stack Integration	Understanding client-server interaction, REST principles, and API communication workflow
3.1.2	Fetching Data from APIs	Using <code>fetch()</code> and <code>axios</code> in React to connect with Spring Boot APIs
3.1.3	Handling Responses and Errors	Displaying server responses, handling 404/500 errors, using AI tools to generate error-handling templates
3.1.4	Authentication and Authorization	Token-based authentication using JWT; AI-assisted code scaffolding for secure endpoints
3.1.5	Connecting MongoDB Data to Frontend	Consuming JSON data from backend and displaying dynamic lists
3.1.6	AI Tools for Integration Debugging	Using Cursor and ChatGPT to identify integration issues and auto-fix code
3.1.7	Mini Project 1 – AI-Supported Task Manager	Build a React + Spring Boot application that manages tasks and syncs data with MongoDB

### In-Class Activities:

- Connect React frontend with a Spring Boot API
- Use Copilot to write Axios fetch logic

#### **Post-Class Assignments:**

- Create a frontend table that dynamically loads backend data
- Implement basic JWT authentication

---

### **Week 2 – Generative AI for Developers**

<b>Lecture</b>	<b>Topic</b>	<b>Description</b>
3.2.1	Introduction to Generative AI	Overview of GenAI evolution, text-to-code, text-to-UI, and AI-driven workflows
3.2.2	Prompt Engineering Essentials	Structure of effective prompts, role prompting, and contextual chaining
3.2.3	Using OpenAI API in Full Stack Apps	Setup, keys, and API calls from React or Node.js
3.2.4	Hugging Face and Gemini API Overview	Integrating open-source models and Google's Gemini API
3.2.5	Image Generation using Stability AI or DALL·E	Building prompt-driven image generation components
3.2.6	Text Summarization and Sentiment Analysis	Using LLMs to analyze user input or summarize feedback
3.2.7	Mini Project 2 – AI Prompt Playground	Build a web app where users can send prompts to OpenAI API and get text/image outputs

#### **In-Class:**

- Connect React form to OpenAI API endpoint
- Test prompts for summarization and rewriting

**Post-Class:**

- Create multiple prompt variations for different outputs
- Document and compare prompt performance

---

## Week 3 – LLM-based App Development

Lecture	Topic	Description
3.3.1	Introduction to LangChain	Concepts of chains, tools, and memory for AI apps
3.3.2	Setting up LangChain.js	Install, initialize, and connect with OpenAI API
3.3.3	Building a Custom Chatbot	Multi-turn conversation with contextual memory
3.3.4	Integrating LangChain with React	Create a simple chat interface that calls LangChain functions
3.3.5	Building Tools and Agents	Add retrieval-based answers using embeddings and vector stores
3.3.6	AI-enhanced Search and Recommendation	Implementing AI-based filtering and suggestions
3.3.7	Mini Project 3 – AI Customer Support Chatbot	Build a LangChain-powered chatbot for a product website with React frontend

**In-Class:**

- Build a basic chatbot using LangChain.js

- Add context persistence using session memory

**Post-Class:**

- Enhance chatbot UI with conversation history
- Document prompts and chain flow

---

## **Week 4 – AI Productivity Tools and Advanced Integrations**

<b>Lecture</b>	<b>Topic</b>	<b>Description</b>
3.4.1	Introduction to AI Developer Tools	Overview: Copilot, Cursor, Codeium, Replit AI, Amazon Q Developer
3.4.2	Using Copilot for Complex Logic	Create backend logic and error handling using Copilot's inline help
3.4.3	Using Cursor for Debugging	Explain errors, generate optimized code snippets, and auto-refactor code
3.4.4	AI for Testing and Automation	Use AI to auto-generate test cases, mocks, and fixtures
3.4.5	Integrating AI in CI/CD	Use GitHub Actions + AI assistants for automating builds
3.4.6	Real-time GenAI Integration	Using WebSockets or Server-Sent Events for AI-generated content updates
3.4.7	Mini Project 4 – AI Debugging & Optimization	Refactor an existing project using Copilot and Cursor to improve performance and maintainability

**In-Class:**

- Demonstrate AI-assisted debugging workflow

- Auto-generate unit tests using AI tools

**Post-Class:**

- Use AI to refactor an existing code block
- Document AI tool impact on productivity

---

**End of Month Project: “GenAI-Powered Knowledge Portal”****Goal:**

Build a full-stack knowledge portal that allows users to ask questions, generate summaries, and view AI-generated insights.

**Key Features:**

- Frontend: React-based UI for prompt submission and display
- Backend: Spring Boot API layer connected to OpenAI API
- AI Layer: LangChain.js for contextual conversation
- Database: MongoDB for storing user queries and responses

**Deliverables:**

- Functional full-stack application
  - API documentation
  - Prompt logs and AI integration notes
  - GitHub repository with instructions
-

# Month 4: Agentic AI + Automation

## Objective:

Enable learners to design and develop autonomous AI agents that can interact with APIs, execute workflows, and perform real-world automation using both **code-based (CrewAI, AutoGen)** and **no-code (Zapier, Chatflow, AgentDK)** tools.

---

## Week 1 – Understanding Agentic AI

Lecture	Topic	Description
4.1.1	Introduction to Agentic AI	Concept of autonomous AI systems; Difference between LLMs and Agents; Real-world use cases
4.1.2	Agent Architecture	Understanding planning, reasoning, and tool usage in AI agents
4.1.3	Components of an Agentic System	Actions, Tools, Memory, Environment; How agents make decisions
4.1.4	LangGraph and CrewAI Overview	Introduction to frameworks that support multi-agent workflows
4.1.5	Setting up Agentic Environment	Installing CrewAI, LangGraph, AutoGen; configuring OpenAI keys
4.1.6	AI Orchestration Concepts	How multiple agents collaborate; task assignment and execution order
4.1.7	Mini Project 1 – Two-Agent Collaboration	Create a “Research Agent” and “Summarizer Agent” that work together to generate topic summaries

## In-Class Activities:

- Create a simple multi-agent setup using CrewAI
- Test agent collaboration using prompts

## Post-Class Assignments:

- Design your own pair of cooperating agents
- Document agent workflow in a diagram

---

## Week 2 – No-Code AI Agents and Workflows

Lecture	Topic	Description
4.2.1	Introduction to No-Code Agent Builders	Overview of platforms like AgentDK, Chatflow AI, Make, and Zapier
4.2.2	AgentDK for Task Automation	Using prompts to create agents that perform specific tasks
4.2.3	Chatflow AI for Conversational Agents	Build and customize a chatbot with workflow logic
4.2.4	Zapier and Make Integrations	Automating workflows between Gmail, Notion, Slack using AI triggers
4.2.5	AI API Chaining in No-Code Tools	Connect multiple APIs through visual flows
4.2.6	Testing and Debugging Agents	Use console logs and AI explainers to fix logic errors
4.2.7	Mini Project 2 – AI Email Summarizer Bot	Create a no-code agent that reads emails and generates daily summaries using OpenAI API

### In-Class:

- Build a Chatflow-based conversational agent
- Connect AI prompt responses to Slack or Google Sheets

### Post-Class:

- Automate a real workflow (e.g., form → summary → notification)
- Share JSON export of no-code agent

---

## Week 3 – Building Intelligent Assistants with CrewAI and AutoGen

Lecture	Topic	Description
4.3.1	Deep Dive into CrewAI Framework	Understanding <code>Agent</code> , <code>Crew</code> , and <code>Task</code> classes
4.3.2	Building Autonomous Workflows	Define step-based task sequences for agents
4.3.3	Using AutoGen Framework	Implement conversational multi-agent collaboration
4.3.4	Adding Tool Access to Agents	Enable agents to call external APIs and perform actions
4.3.5	Integrating Memory	Store agent context and past results using vector databases
4.3.6	Combining Agents with Frontend	Create a React UI that interacts with multi-agent backend
4.3.7	Mini Project 3 – AI Research Crew	Build a CrewAI system with “Searcher”, “Analyst”, and “Presenter” agents that collaborate to produce insights

### In-Class:

- Setup and run a CrewAI-based task sequence
- Add API access tools for your agents

### Post-Class:

- Implement memory layer using JSON or MongoDB

- Extend CrewAI project with one additional agent

---

## Week 4 – AI Automation and Integration

Lecture	Topic	Description
4.4.1	Introduction to AI Automation	Understanding end-to-end automation pipelines with LLMs
4.4.2	Agent Scheduling and Triggers	Use CRON jobs and webhook triggers for autonomous execution
4.4.3	Combining No-Code + Code Agents	Hybrid approach: CrewAI backend with Zapier front-end automation
4.4.4	Real-Time Decision Making	Build agents that use reasoning to pick actions dynamically
4.4.5	Workflow Optimization	Using AI to identify redundant steps and improve performance
4.4.6	Monitoring and Logs	Logging AI decisions, tracking success/failure metrics
4.4.7	Mini Project 4 – AI Task Orchestrator	Create an automated workflow where multiple agents handle research, summarization, and email delivery

### In-Class:

- Demonstrate hybrid automation (CrewAI + Zapier)
- Set up automated email updates from an AI system

### Post-Class:

- Document automation sequence diagram
- Compare performance between AI-driven vs manual workflows

---

## End of Month Project: “Autonomous Research and Reporting Agent System”

### Goal:

Build a complete Agentic AI system that autonomously gathers, analyzes, and reports information on any topic.

### Key Features:

- Multi-agent workflow (Researcher, Writer, Editor, Reporter)
- Uses OpenAI API or Gemini for content generation
- CrewAI or AutoGen framework for orchestration
- Optional no-code integration with Zapier or Chatflow for scheduling and delivery

### Deliverables:

- Functional autonomous system
- Workflow diagram and architecture explanation
- Source code (GitHub) and live demo
- Documentation of agent behavior and roles

---

## Month 5: Cloud + DevOps + GenAI Deployment

### Objective:

Enable learners to understand cloud fundamentals, deploy full-stack applications (with AI components), and automate their build, test, and deployment workflows using AI-powered DevOps tools and CI/CD pipelines.

---

### Week 1 – Cloud Fundamentals and Architecture

Lecture	Topic	Description
5.1.1	Introduction to Cloud Computing	Cloud concepts (IaaS, PaaS, SaaS), deployment models (public, private, hybrid)
5.1.2	Understanding Modern Cloud Architecture	Three-tier architecture, microservices, serverless model
5.1.3	Overview of Cloud Platforms	AWS, Google Cloud, Azure, Firebase, Render, Vercel
5.1.4	AI-Assisted Cloud Setup	Use AI tools (ChatGPT, Copilot) to create configuration templates for Vercel/Render
5.1.5	Hosting Static Frontend Apps	Deploy React apps on Vercel or Netlify
5.1.6	Hosting Spring Boot Backend	Deploy backend on Render or Railway with AI-generated YAML configurations
5.1.7	Mini Project 1 – Deploy a Full Stack App	Deploy React + Spring Boot + MongoDB application using Vercel and Render

#### **In-Class Activities:**

- Deploy a React project live on Vercel
- Configure environment variables using AI-suggested steps

#### **Post-Class Assignments:**

- Deploy your backend API on Render or Railway
- Create a documentation guide using AI summarization

---

## Week 2 – AI-Assisted Deployment and Scaling

Lecture	Topic	Description
5.2.1	Cloud Deployment Lifecycle	Build → Package → Deploy → Monitor process explained
5.2.2	Docker for Containerization	Creating Dockerfiles; AI-assisted Dockerfile generation
5.2.3	Multi-Container Applications	Docker Compose setup for full stack projects
5.2.4	AI Tools for Deployment	Using Copilot and ChatGPT for generating cloud config scripts
5.2.5	Auto-Scaling and Load Balancing	Understanding horizontal vs vertical scaling
5.2.6	Cloud Cost Optimization	Use AI estimators to predict and reduce cost usage
5.2.7	Mini Project 2 – Containerized AI App	Dockerize and deploy a GenAI application with frontend, backend, and AI API integration

### In-Class:

- Generate Dockerfile with Copilot
- Deploy containerized app on Render or Railway

### Post-Class:

- Add Docker Compose for database + backend setup

- Document deployment workflow

---

## Week 3 – CI/CD Pipelines and AI Integration

Lecture	Topic	Description
5.3.1	Introduction to DevOps and CI/CD	Continuous Integration, Continuous Deployment concepts
5.3.2	GitHub Actions Overview	Creating workflows for automated build and deploy
5.3.3	AI-Assisted CI/CD Setup	Using Copilot to generate YAML workflows for React + Spring Boot apps
5.3.4	Integrating Testing Pipelines	Add automated tests with Jest, JUnit; AI-generated test scripts
5.3.5	Monitoring and Alerts	Use GitHub Actions, Slack bots, and AI summarizers for status updates
5.3.6	AI for Code Review and Merging	Auto-review pull requests with Copilot or ChatGPT plugins
5.3.7	Mini Project 3 – Automated Build + Deploy	Set up a pipeline that builds, tests, and deploys a GenAI app automatically to Vercel and Render

### In-Class:

- Configure GitHub Action with AI help
- Test CI/CD pipeline execution live

### Post-Class:

- Add unit testing automation
- Share GitHub workflow YAML file

---

## Week 4 – AI-Powered Cloud Deployment and Observability

Lecture	Topic	Description
5.4.1	Integrating AI Models in Production	Deploying OpenAI and Hugging Face APIs securely
5.4.2	Serverless Functions and Edge AI	Using Vercel Serverless and Firebase Functions for lightweight GenAI apps
5.4.3	Real-Time Analytics with AI	Using LangFuse or LlamaIndex for prompt logging and analysis
5.4.4	Observability and Monitoring	Using tools like Grafana, Prometheus; AI-based anomaly detection
5.4.5	Security Best Practices	Managing API keys, environment variables, and access control
5.4.6	Generative AI Ops (GenOps)	Using AI to predict downtime, analyze usage, and auto-scale
5.4.7	Mini Project 4 – Deploy an AI Chatbot to Cloud	Deploy an LLM-powered chatbot using serverless backend and real-time monitoring dashboard

### In-Class:

- Configure API key management securely

- Enable serverless deployment of AI chatbot

**Post-Class:**

- Implement prompt logging using LangFuse
- Analyze app logs with AI-based explainers

---

**End of Month Project: “Cloud-Deployed GenAI Platform”****Goal:**

Build and deploy a full-scale GenAI application with cloud hosting, CI/CD automation, and observability enabled.

**Key Features:**

- Frontend: React with Tailwind (Vercel)
- Backend: Spring Boot (Render or Railway)
- Database: MongoDB (Atlas)
- AI Layer: OpenAI or Gemini API
- CI/CD: GitHub Actions with automated testing and deployment
- Monitoring: Grafana or AI-assisted dashboards

**Deliverables:**

- Hosted app link (frontend + backend)
- GitHub repository with CI/CD pipeline
- Deployment documentation (generated using AI tools)

- Observability dashboard screenshots

---

## Month 6: Capstone + Industry Readiness

### Objective:

Empower learners to build an industry-standard AI-integrated product, automate testing and deployment, and prepare them for placement opportunities through AI-assisted resume review, portfolio building, and mock interviews.

---

### Week 1 – Capstone Project Planning and Architecture

Lecture	Topic	Description
6.1.1	Capstone Overview and Evaluation Criteria	Understanding end-to-end expectations: architecture, innovation, AI integration, scalability, and presentation
6.1.2	Ideation and Problem Scoping	Use AI (ChatGPT, Gemini) to brainstorm project ideas and validate feasibility
6.1.3	Architecture Design	Use AI tools like Diagrams.net or Draw.io (with ChatGPT prompts) to design system architecture
6.1.4	Defining Modules and Tech Stack	Choose between React, Spring Boot, MongoDB, and GenAI APIs; Define modules and dependencies
6.1.5	Wireframing and UI Design	Use Figma AI or Galileo AI for prototype generation
6.1.6	Documentation Framework	Use AI to auto-generate project documentation (Problem statement, Architecture, User flow)
6.1.7	Mini Project 1 – Project Blueprint	Submit a project proposal document including system flow, tech stack, and design mockups

### In-Class Activities:

- AI-assisted brainstorming sessions

- Architecture diagram building with ChatGPT guidance

#### **Post-Class Assignments:**

- Write project summary and module breakdown using AI tools
- Create Figma-based prototype for approval

---

### **Week 2 – Implementation Phase (Frontend + Backend)**

<b>Lecture</b>	<b>Topic</b>	<b>Description</b>
6.2.1	Frontend Setup	Initialize React + Tailwind project; integrate AI-generated UI components
6.2.2	Backend Setup	Configure Spring Boot backend with required models and controllers
6.2.3	API Development	Build REST endpoints and connect to MongoDB
6.2.4	Integrating GenAI APIs	Add OpenAI, Gemini, or Hugging Face API features
6.2.5	Agentic Integration (Optional Advanced)	Add a CrewAI or AutoGen agent for task automation
6.2.6	Connecting Frontend and Backend	Build full-stack communication with Axios and JWT auth
6.2.7	Mini Project 2 – AI-Powered Module Demo	Deliver one working AI-driven feature (e.g., summarizer, chatbot, recommender)

#### **In-Class:**

- Real-time debugging with Copilot and Cursor
- API testing using Postman AI

**Post-Class:**

- Complete module integration
- Push working version to GitHub

---

**Week 3 – Testing, CI/CD, and Deployment**

Lecture	Topic	Description
6.3.1	Unit Testing with AI	Generate test cases using Copilot and ChatGPT
6.3.2	Frontend Testing	Write Jest and React Testing Library scripts with AI suggestions
6.3.3	Backend Testing	Use JUnit and Postman AI for API validation
6.3.4	Integration Testing	Verify end-to-end flow using automated test cases
6.3.5	CI/CD Implementation	Set up GitHub Actions for build and deploy automation
6.3.6	Deployment	Host the complete system (frontend + backend) on Vercel, Render, or Railway
6.3.7	Mini Project 3 – Full Stack Deployment	Deploy your project to live environment with automated testing pipeline

**In-Class:**

- Configure automated pipeline in GitHub Actions
- Test cloud deployment steps live

**Post-Class:**

- Finalize hosted app deployment

- Share live demo link with mentors

---

## Week 4 – Presentation, Evaluation, and Industry Readiness

Lecture	Topic	Description
6.4.1	Project Documentation and README	Auto-generate documentation and README using AI summarization
6.4.2	Creating Project Demo Video	Record walkthrough using Loom or Vidyard; write narration script using ChatGPT
6.4.3	Portfolio Building	Publish project on GitHub, Netlify/Vercel; Showcase on LinkedIn with AI-written description
6.4.4	Resume and Profile Optimization	Use AI tools (Rezi, Teal, ChatGPT) to enhance resume and GitHub profile
6.4.5	Mock Interview Prep	Practice AI-assisted mock interviews using Interview Warmup and ChatGPT
6.4.6	Project Evaluation Rubrics	Evaluation on technical depth, AI integration, code quality, deployment, and presentation
6.4.7	Final Project Showcase	Learners present their projects to mentors; Peer and instructor reviews

### In-Class:

- AI-assisted resume improvement session
- Conduct mock project pitch simulation

### Post-Class:

- Publish GitHub project with full documentation

- Share project post on LinkedIn

---

## End of Program Capstone: “AI-Integrated Full Stack Product Launch”

### Goal:

Build, deploy, and present a scalable AI-driven full-stack product with real-world relevance.

### Examples:

- **AI Resume Analyzer:** Upload resume, get feedback from LLM
- **Smart Task Tracker:** GenAI-summarized progress reports
- **E-Learning Portal:** AI quiz generator and chatbot instructor
- **News Summarization Dashboard:** Fetches and summarizes daily content

### Key Deliverables:

- Live deployed project (frontend + backend + AI integration)
- Source code (GitHub)
- Demo video walkthrough
- Documentation (AI-generated where applicable)
- Resume + Portfolio + LinkedIn post

### Evaluation Parameters:

Area	Weightage	Description
Technical Implementation	30%	Code quality, modularity, performance
AI Integration	20%	Proper and creative use of AI APIs or agents

Deployment & CI/CD	20%	Working live system with automated deployment
Documentation & Presentation	20%	Clear, complete documentation and demo
Communication	10%	Clarity in presenting project and answering queries

---