

# Course Overview



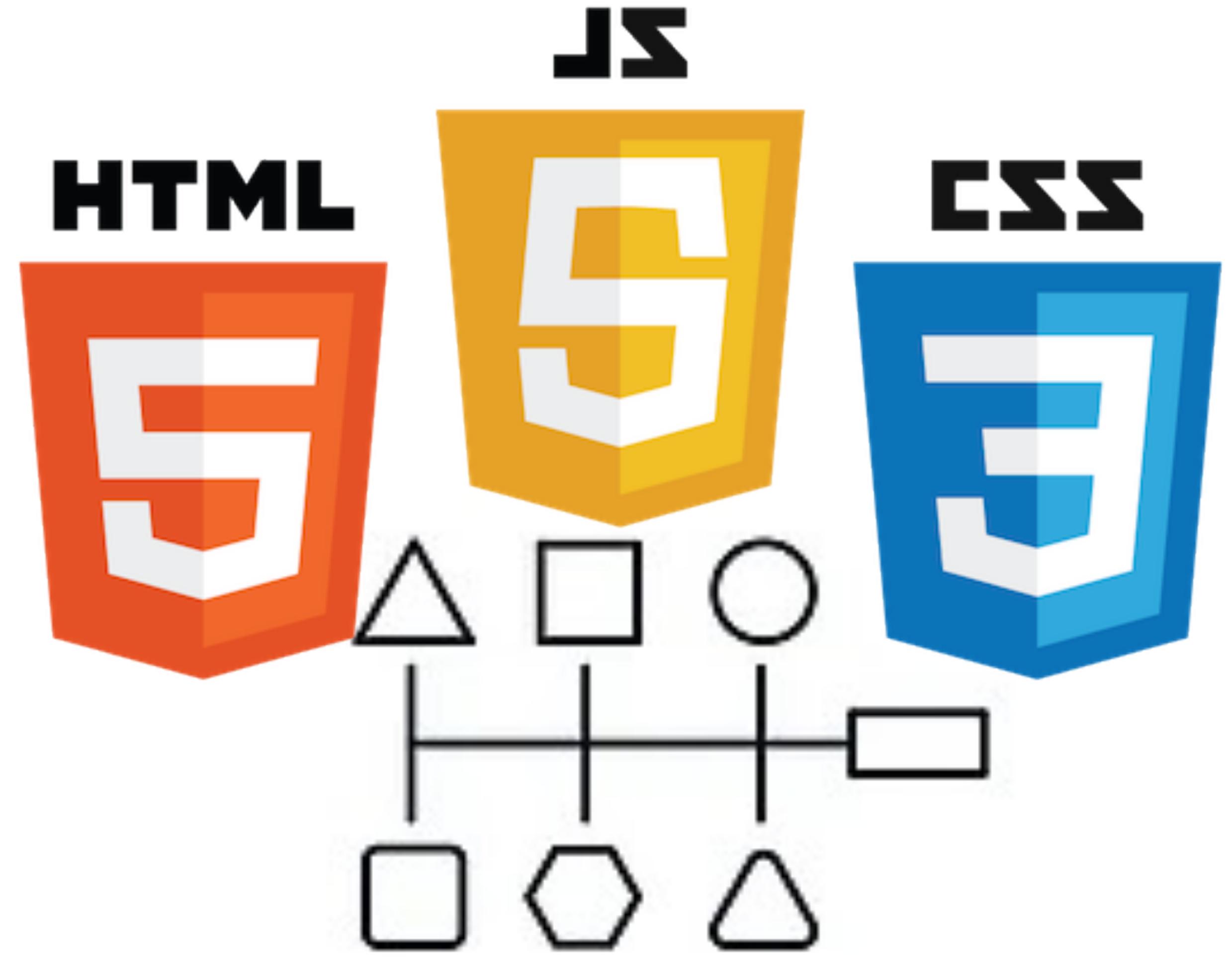
Full Stack Development 1

# Agenda

- Prerequisites
- Module Overview / Calendar
- Assessment Schedule
- Anatomy of a Topic
- References

# Perquisites

- Foundation level skills in:
  - **HTML:** Ability to effectively structure the html content of a small to medium static site, including some understanding of the use of templates
  - **CSS:** Understand the fundamentals of CSS, and be able to realise simple layouts and designs. Be comfortably using a CSS framework



- **Programming:** Intermediate level programming programming ability in Javascript or any C like language (Java, C#, Go etc)

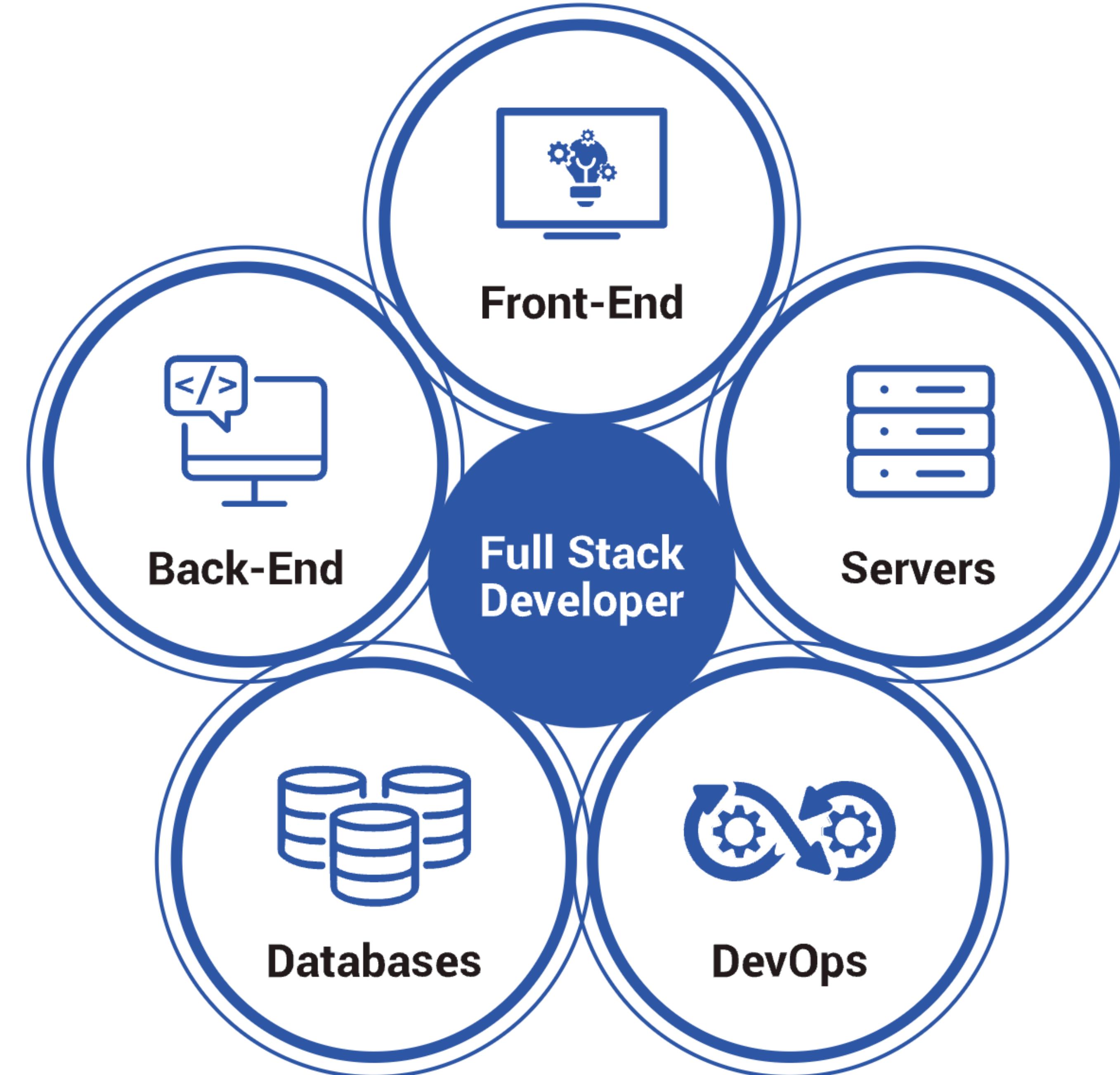
Back End Web  
Development

Test Driven  
Development

Application  
Programmer  
Interfaces

Front End Web  
Development

Full Stack



# Back End Web Development

Test Driven Development

Application Programmer Interfaces

Front End Web Development

Full Stack

The image shows a horizontal row of four cards, each representing a different component of a back-end web development stack. The cards are light green with rounded corners and a thin black border. Each card features a large icon on the left and a title at the top.

- 01: Node** (Icon: Node.js logo) - The fundamentals of Node. A first Node application.
- 02: Hapi.js** (Icon: Hapi.js logo) - The Hapi application Framework
- 03: Joi** (Icon: Spatula icon) - Validation with Joi
- 05: Models** (Icon: Folder with cubes icon) - Incorporate Mongo Models

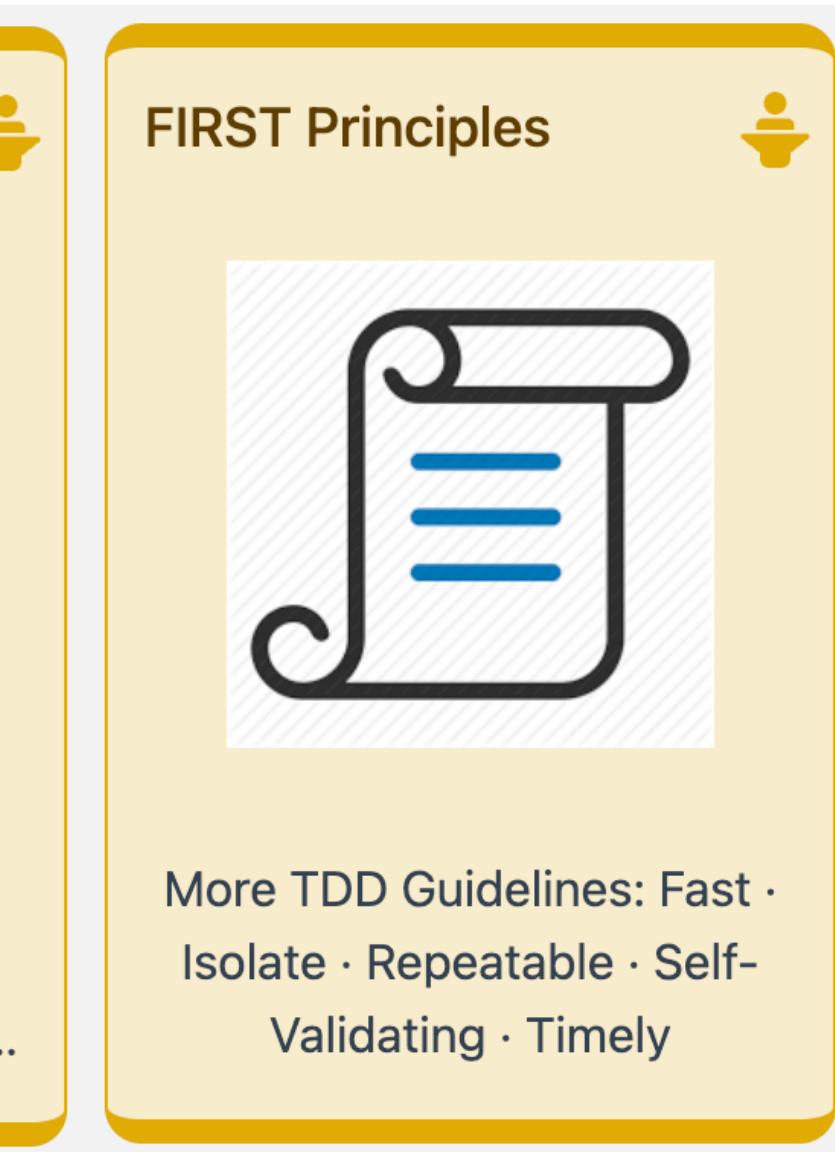
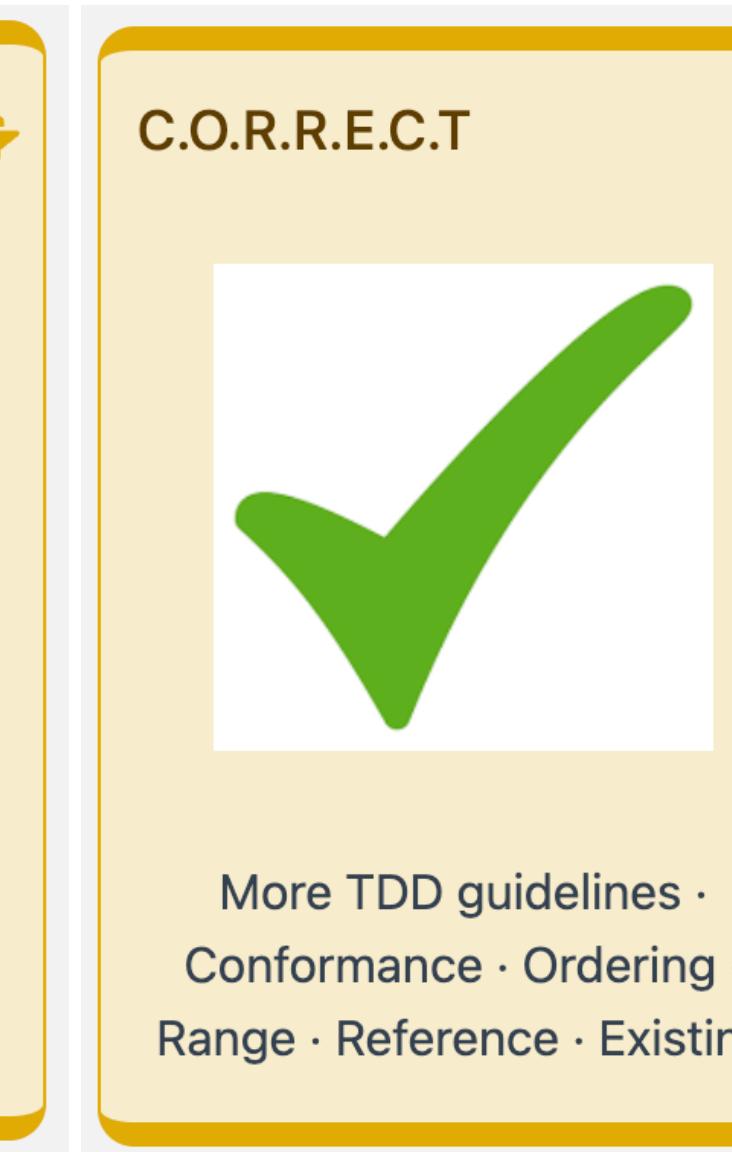
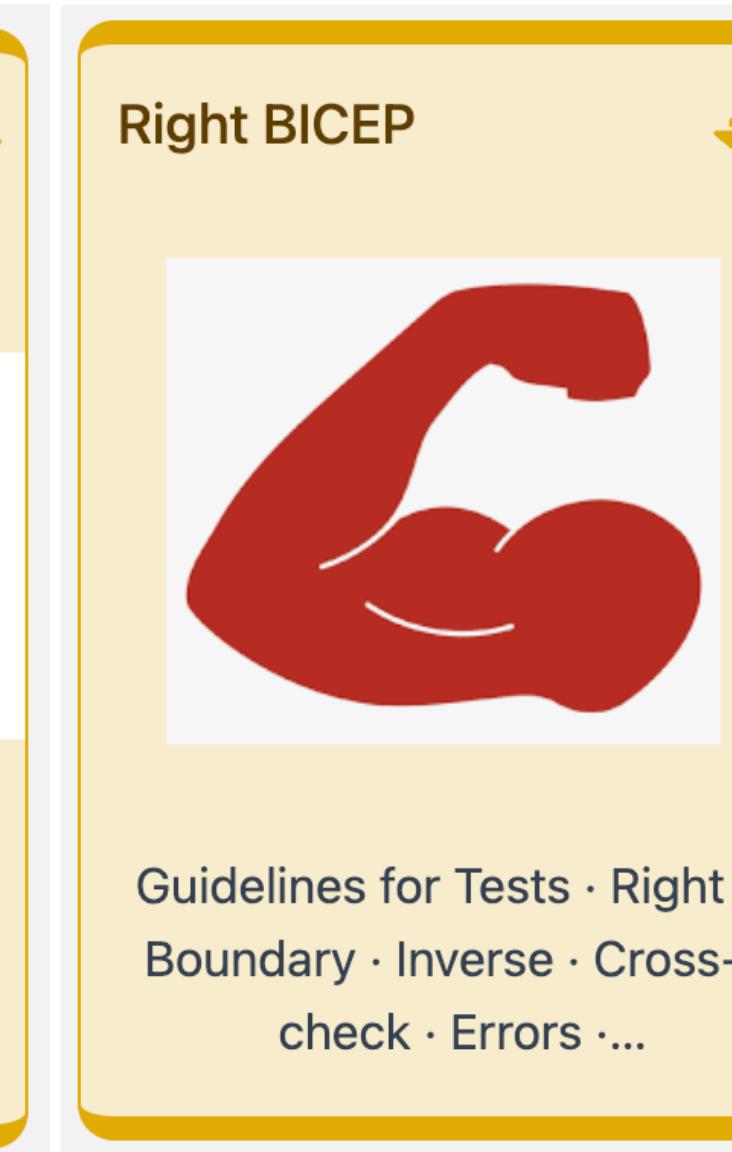
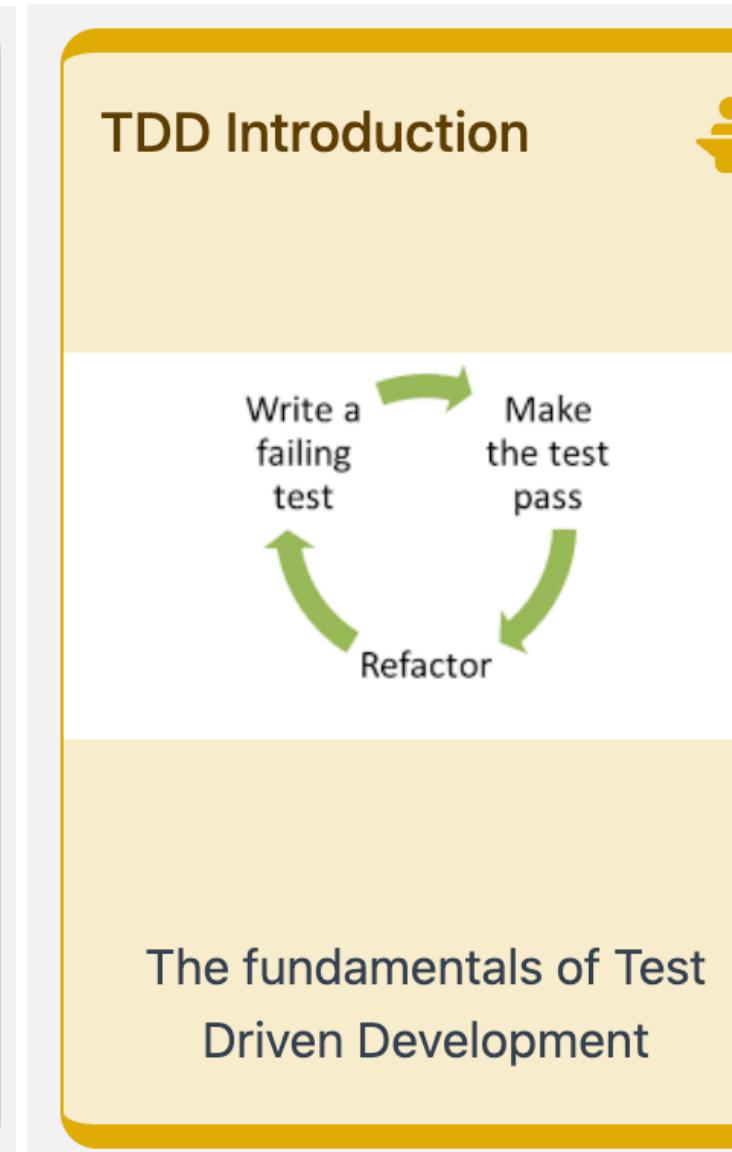
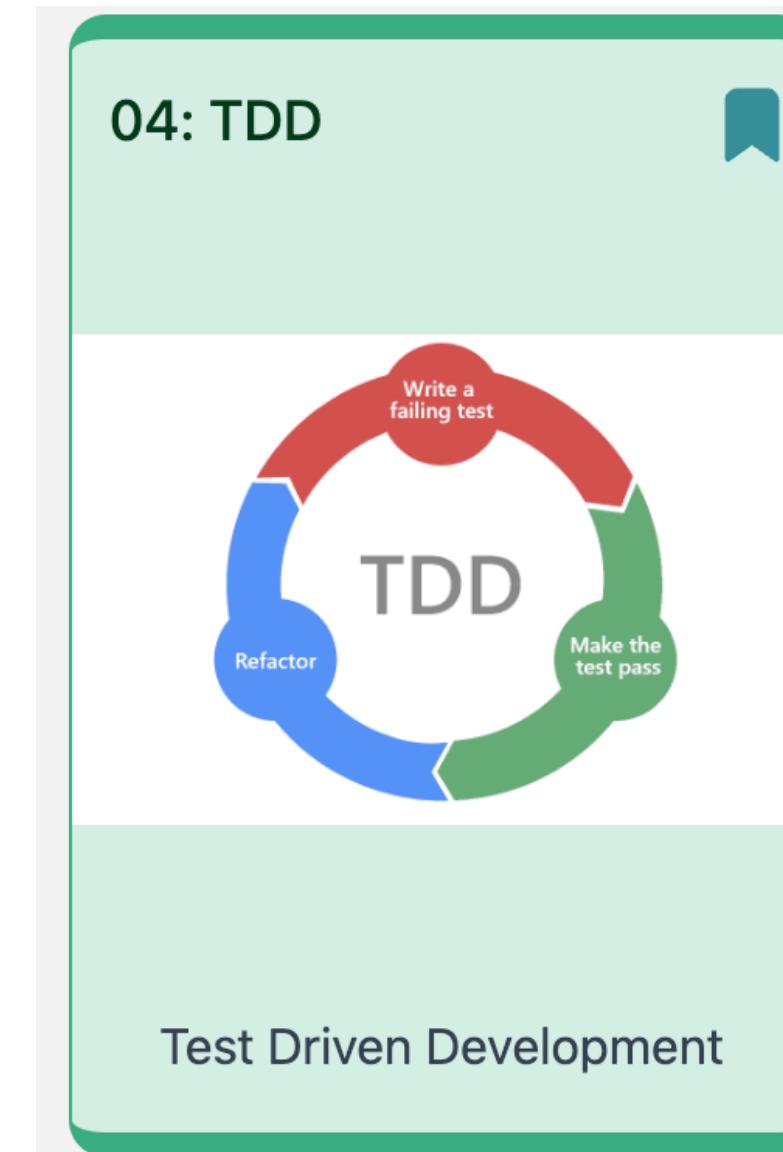
# Back End Web Development

## **Test Driven Development**

Application  
Programmer  
Interfaces

# Front End Web Development

Full Stack



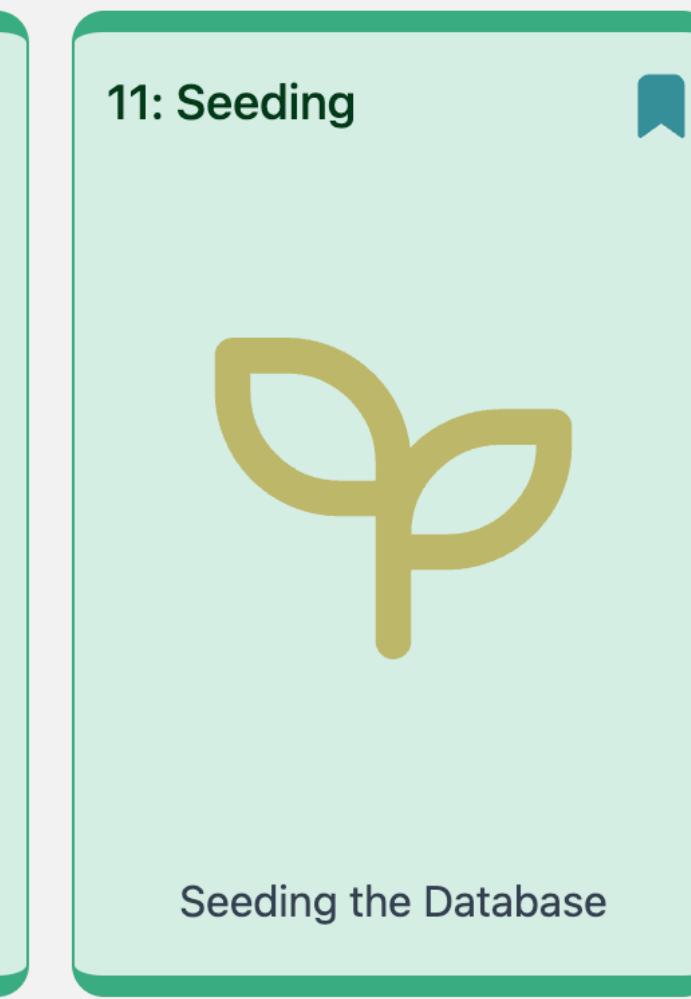
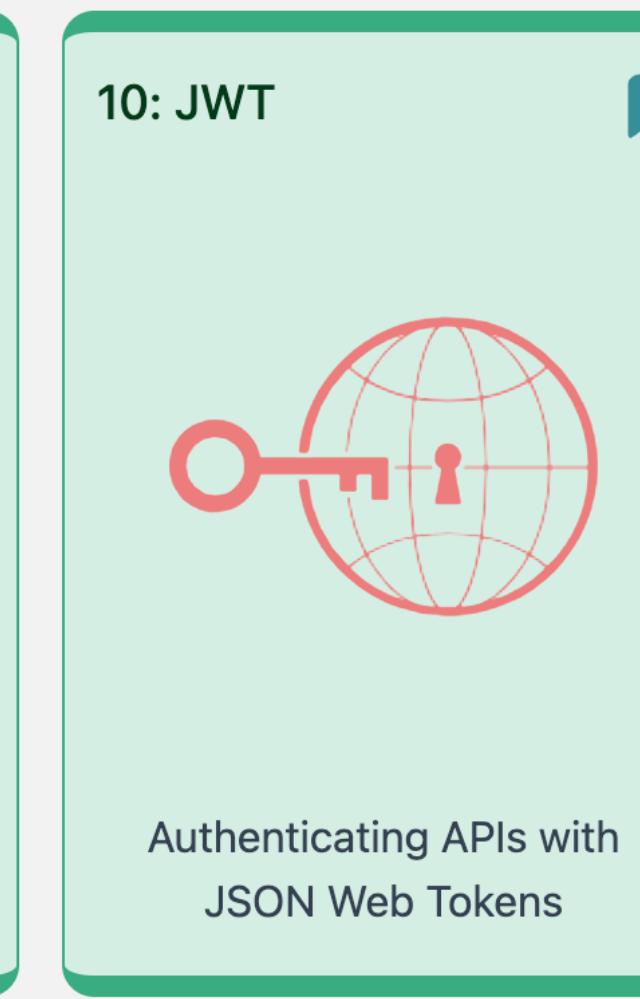
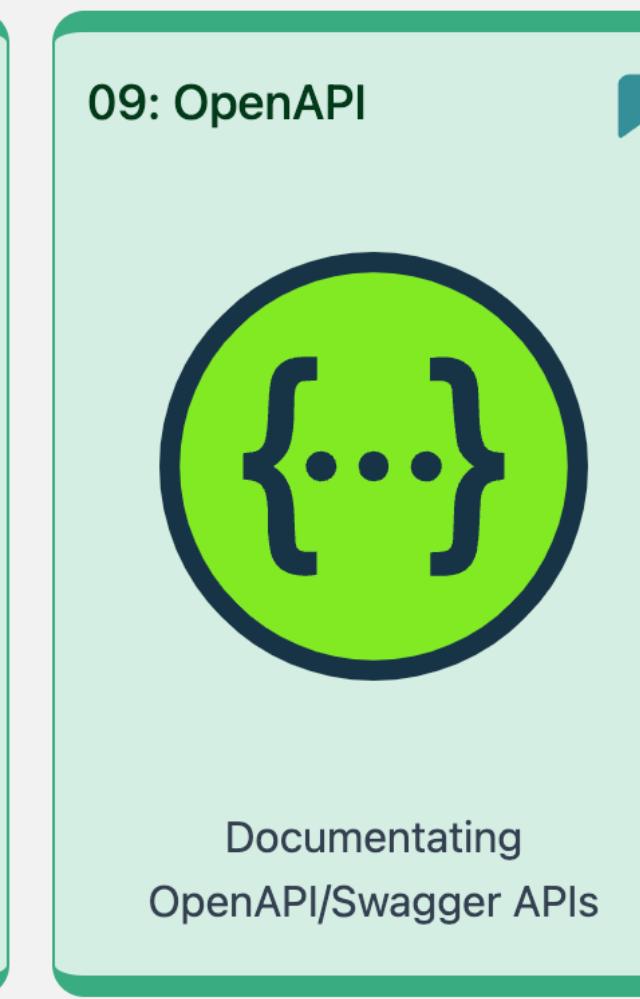
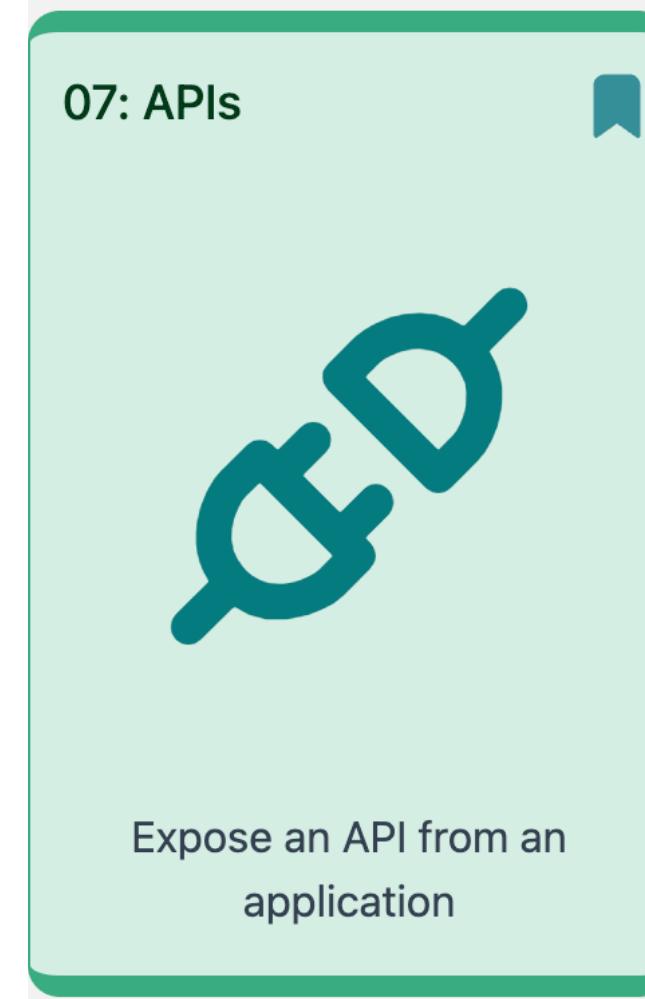
Back End Web  
Development

Test Driven  
Development

## **Application Programmer Interfaces**

Front End Web  
Development

Full Stack



Back End Web  
Development

Test Driven  
Development

Application  
Programmer  
Interfaces

**Front End Web  
Development**

Full Stack

The image shows a horizontal row of seven cards, each representing a lesson in the Svelte framework. Each card has a green header with a bookmark icon, a large central icon, and a brief description below it.

- 12: Deployment** (Icon: Cloud with arrow): Deploying to Glitch, Render & Cloud Atlas
- 13: DOM** (Icon: Document tree): The Document Object Model
- 14: Svelte Introduction** (Icon: Large orange Svelte logo): First contact with the Svelte Framework
- 15: Svelte Components** (Icon: Component diagram): Introducing the Svelte Component Model
- 17: Donation Applications** (Icon: Computer monitor with gears): Overview of Donation Applications - Hapi & Svelte
- 18: TypeScript** (Icon: TypeScript logo): TypeScript Hapi Applications
- 19: SvelteKit Routing** (Icon: Route sign): Application Routing in SvelteKit

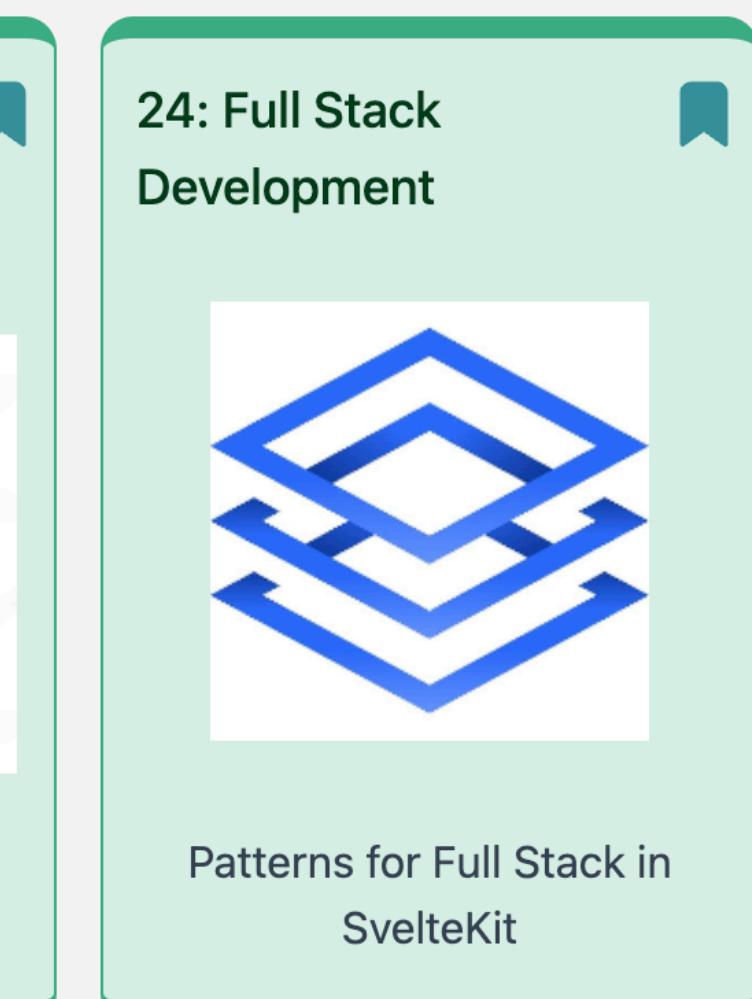
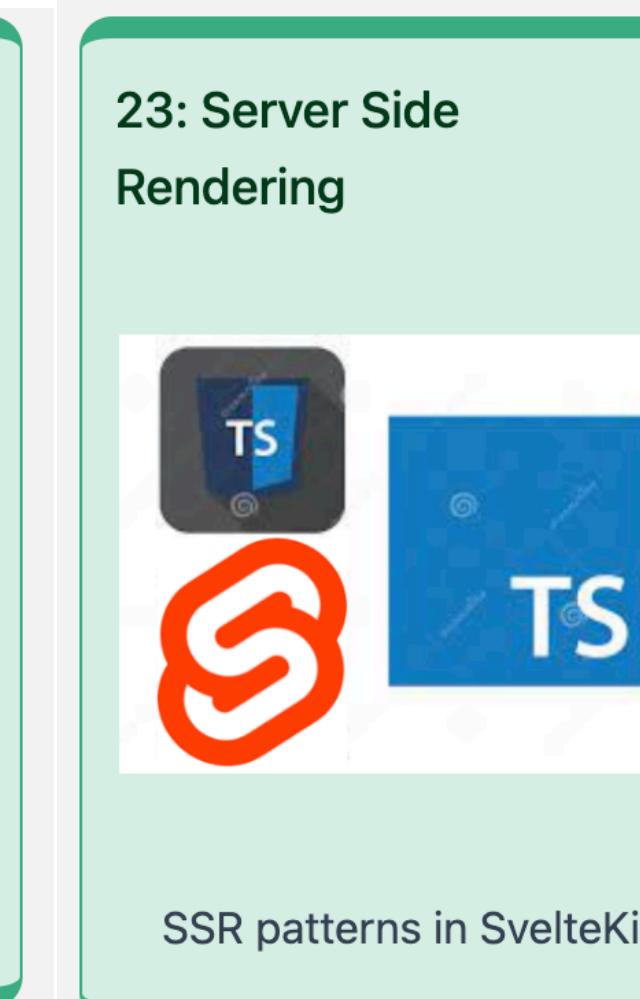
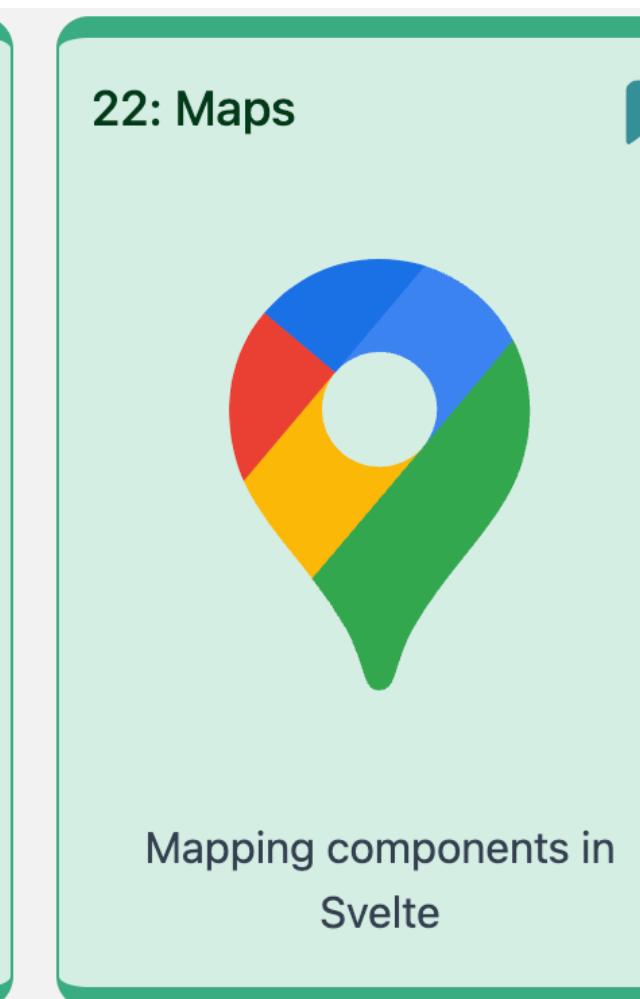
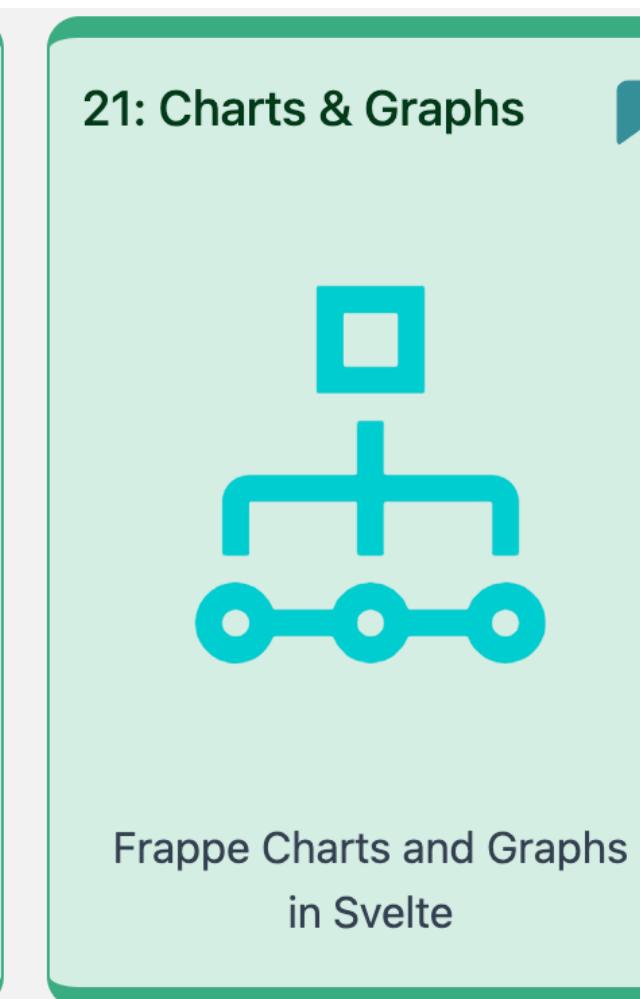
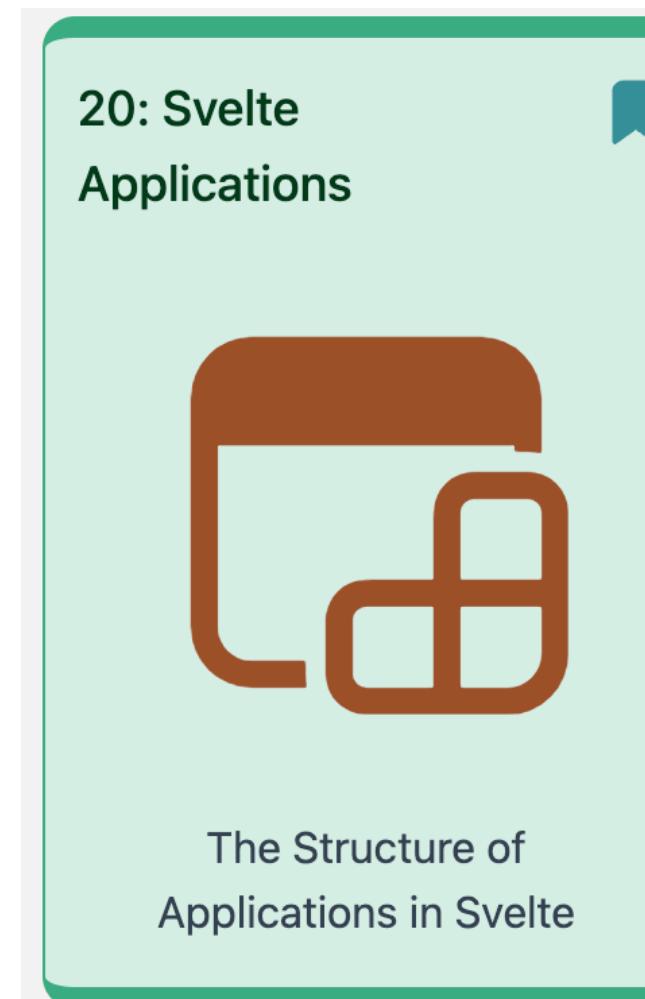
Back End Web  
Development

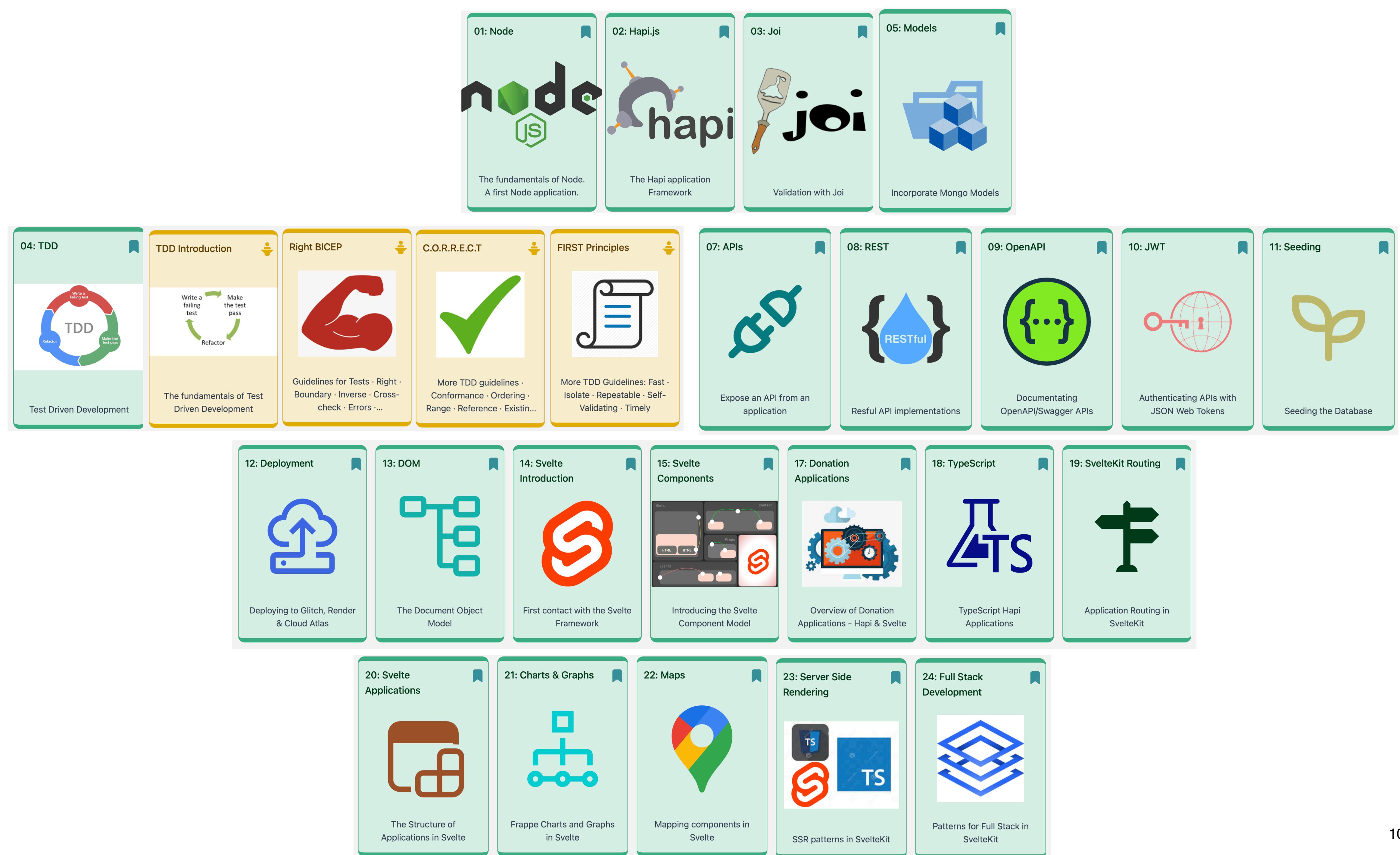
Test Driven  
Development

Application  
Programmer  
Interfaces

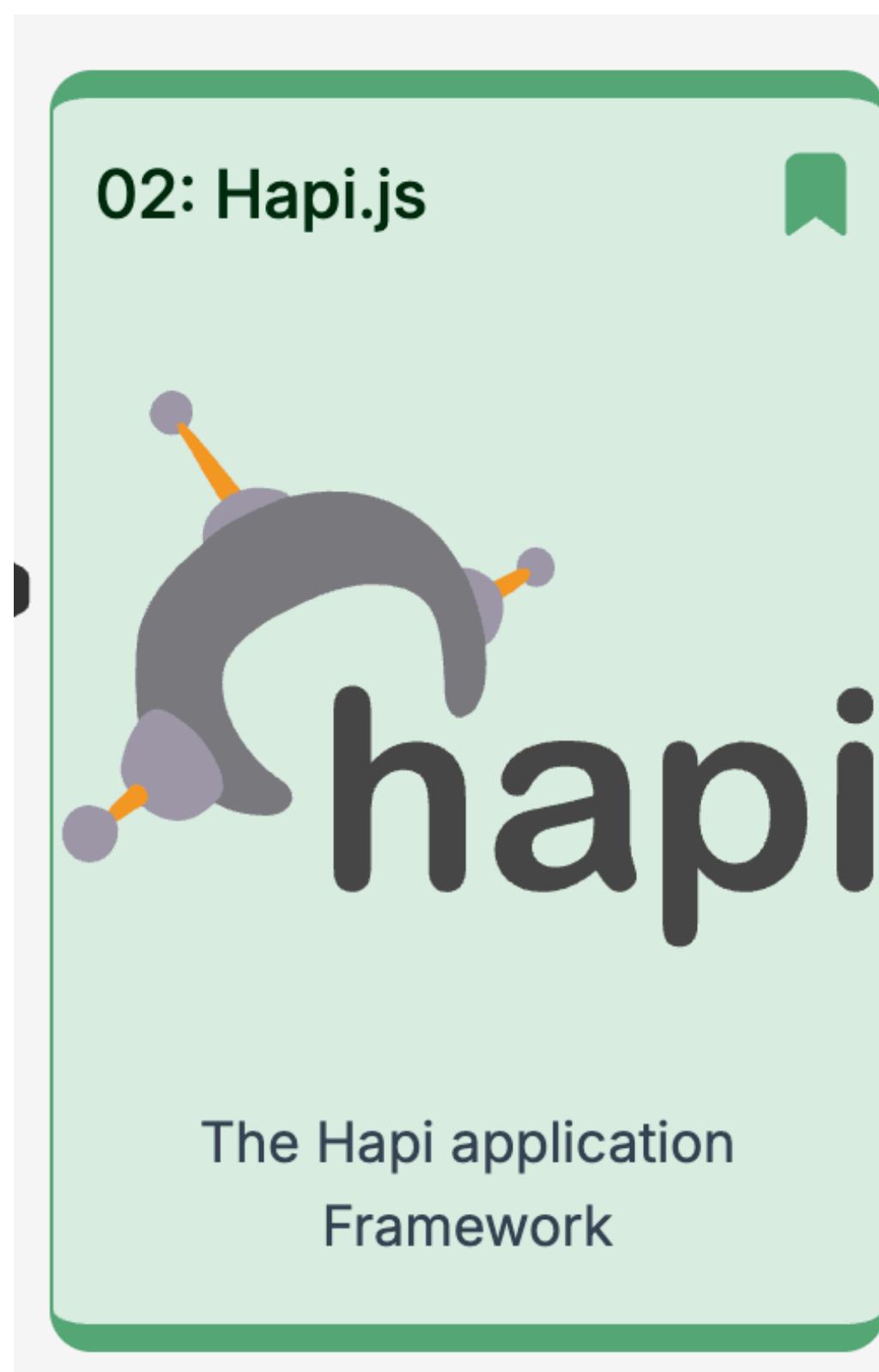
Front End Web  
Development

**Full Stack**

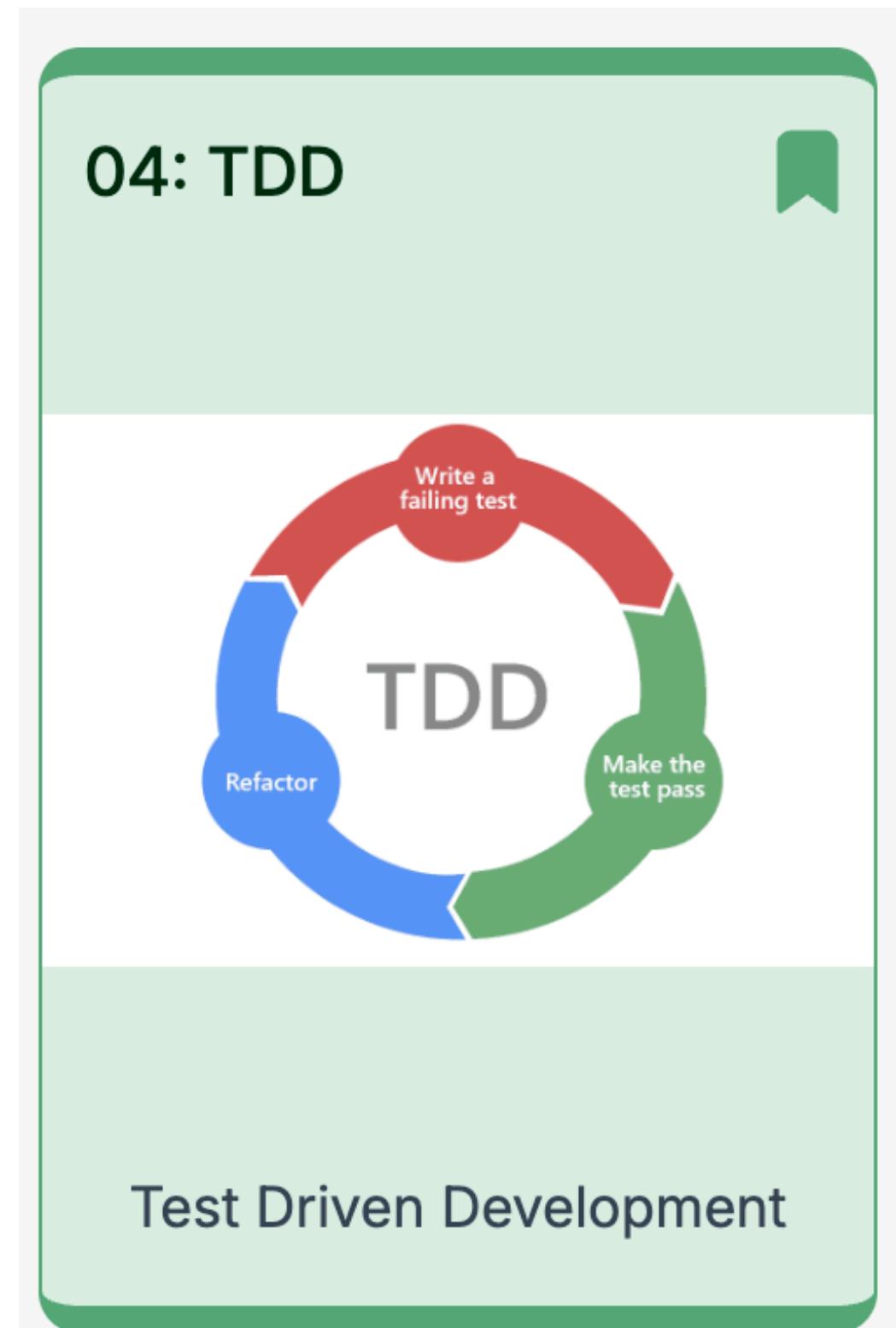
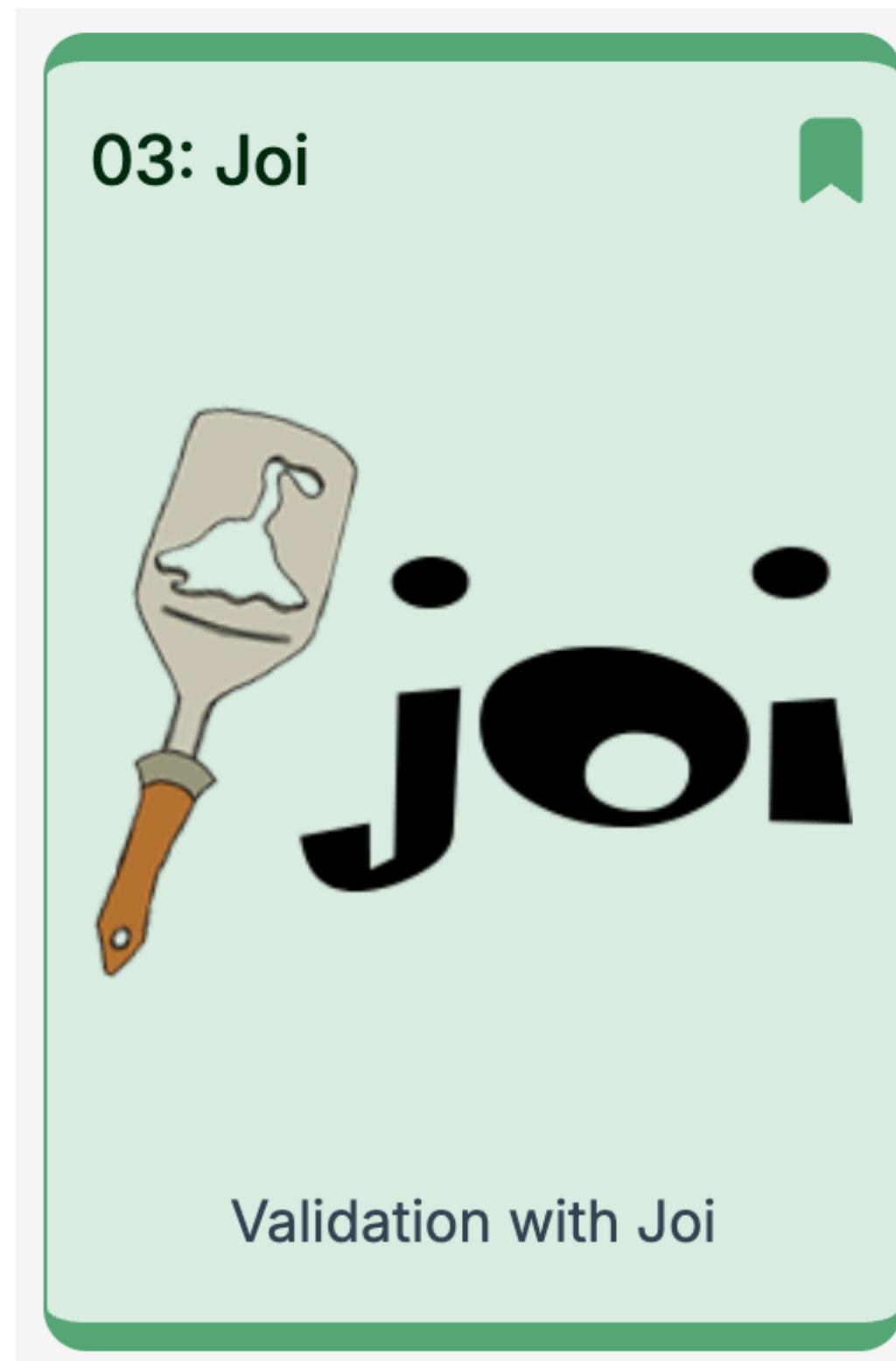




		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22



		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22



		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

## 05: Models



Incorporate Mongo Models

## 06: Assignment 1



Concept, detail and grading rubric

		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22



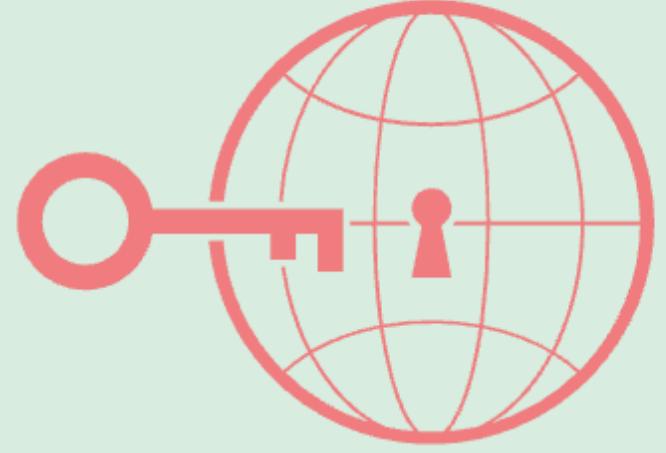
		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

**09: OpenAPI**



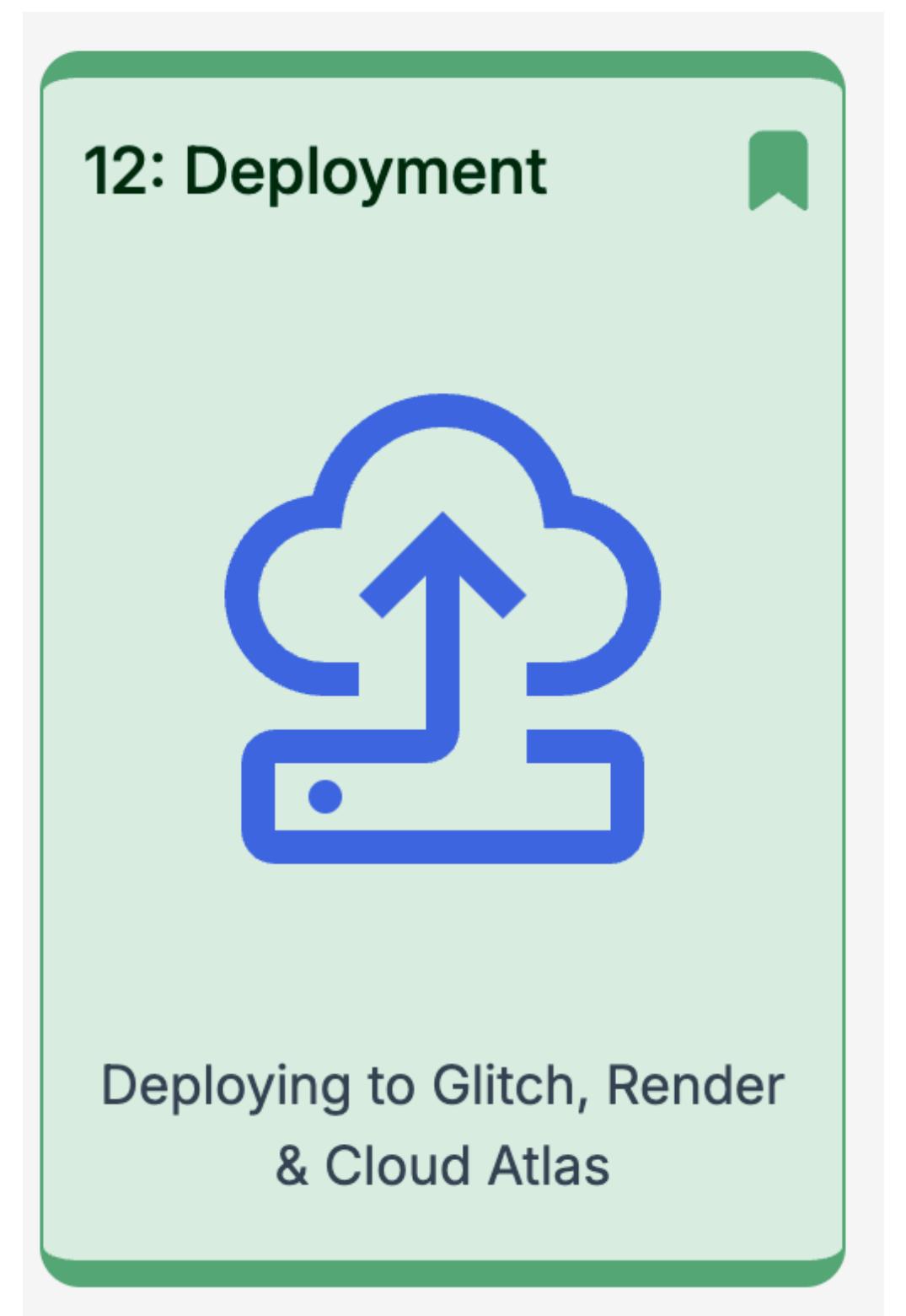
Documentating  
OpenAPI/Swagger APIs

**10: JWT**



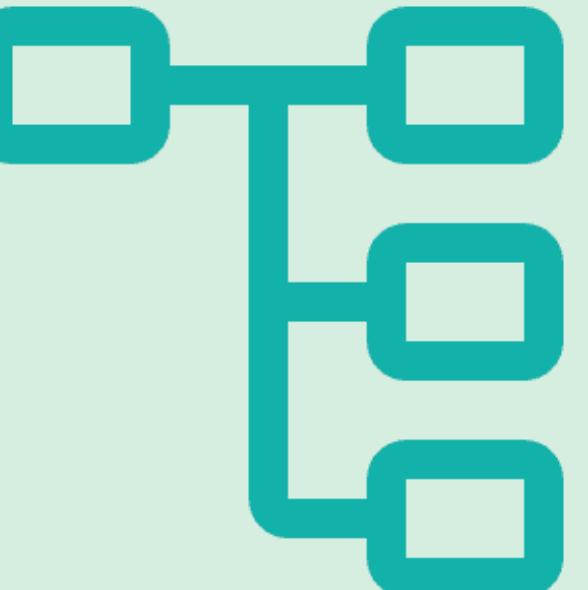
Authenticating APIs with  
JSON Web Tokens

		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22



		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

**13: DOM**



The Document Object Model  
Model

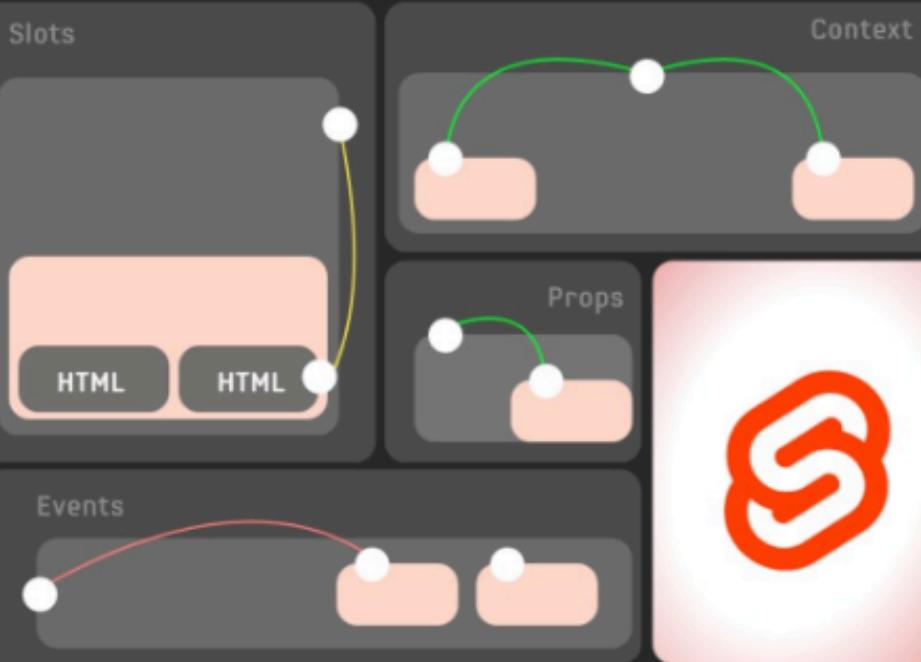
**14: Svelte  
Introduction**



First encounters with the  
Svelte Framework

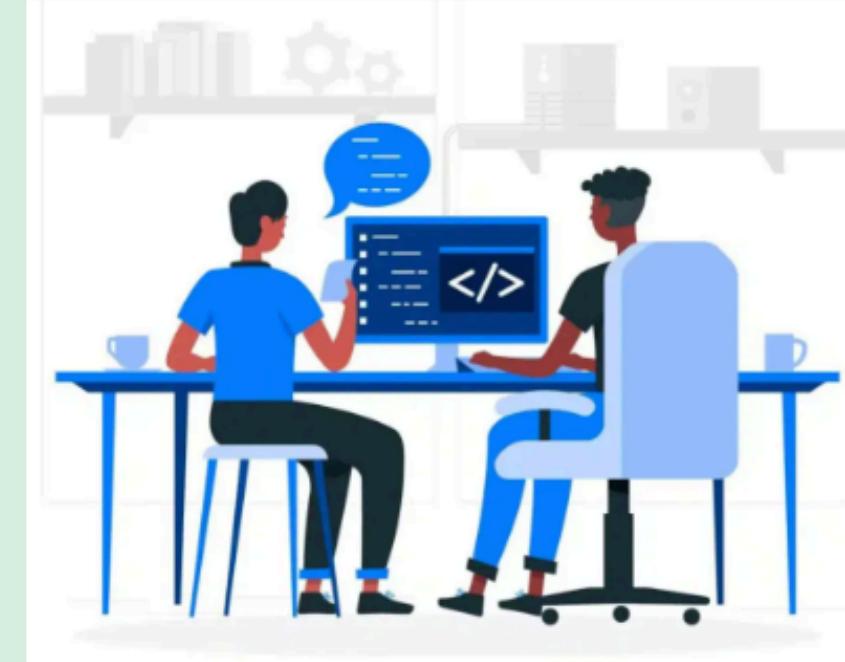
		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

## 15. Svelte Components



Introducing the Svelte Component Model

## 16: Assignment 2



Concept, detail and grading rubric

		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

## 17: Donation Applications



Overview of Donation Applications - Hapi & Svelte

## 18: TypeScript



TypeScript Hapi Applications

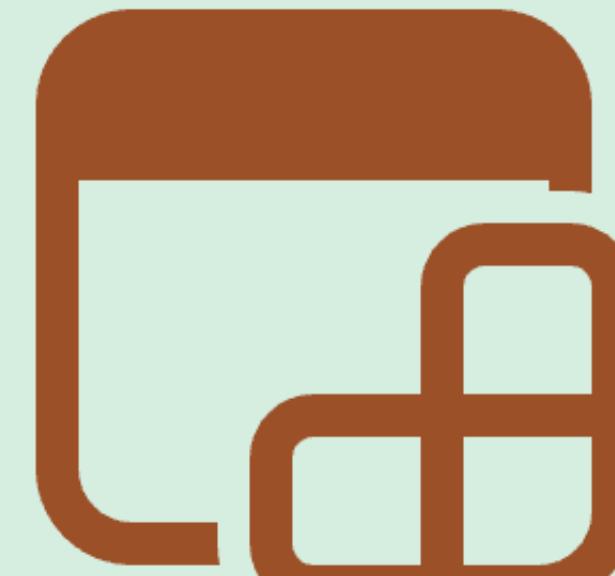
		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
	EASTER 1	29	30	31	1	2	3
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

## 19: SvelteKit Routing



Application Routing in SvelteKit

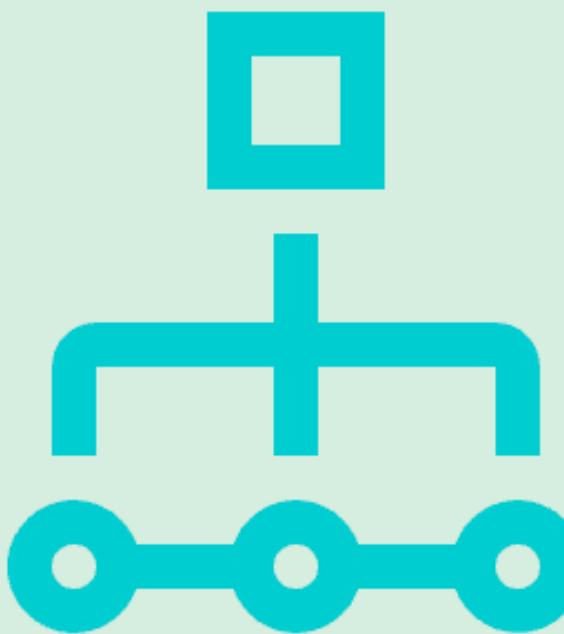
## 20: Svelte Applications



The Structure of Applications in Svelte

		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
EASTER 1	29	30	31	1	2	3	
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

## 21: Charts & Graphs



Frappe Charts and Graphs in Svelte

## 22: Maps



Mapping components in Svelte

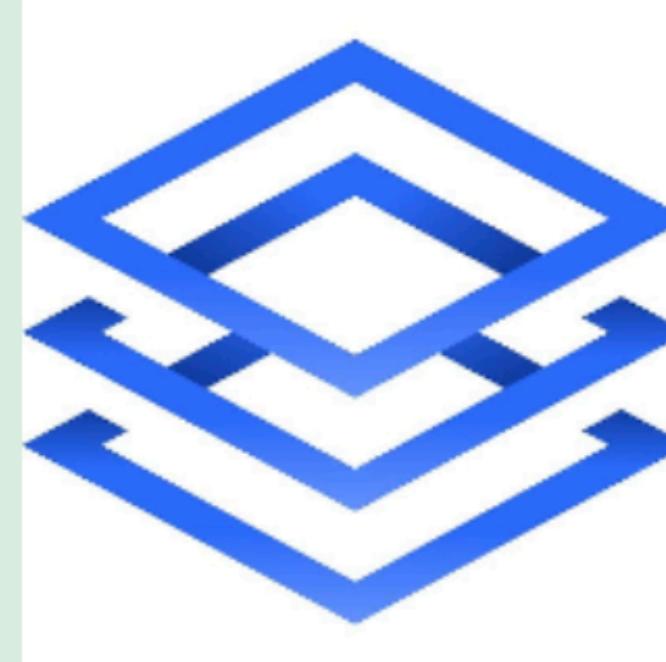
		S	M	T	W	T	F
JAN	WEEK	11	12	13	14	15	16
	1	18	19	20	21	22	23
	2	25	26	27	28	29	30
FEB	3	1	2	3	4	5	6
	4	8	9	10	11	12	13
	READING	15	16	17	18	19	20
	5	22	23	24	25	26	27
MAR	6	1	2	3	4	5	6
	READING	8	9	10	11	12	13
	7	15	16	17	18	19	20
	8	22	23	24	25	26	27
EASTER 1	29	30	31	1	2	3	
APR	EASTER 2	5	6	7	8	9	10
	9	12	13	14	15	16	17
	10	19	20	21	22	23	24
MAY	11	26	27	28	29	30	31
	12	3	4	5	6	7	8
	EXAM	10	11	12	13	14	15
	EXAM	17	18	19	20	21	22

**23: Server Side Rendering**

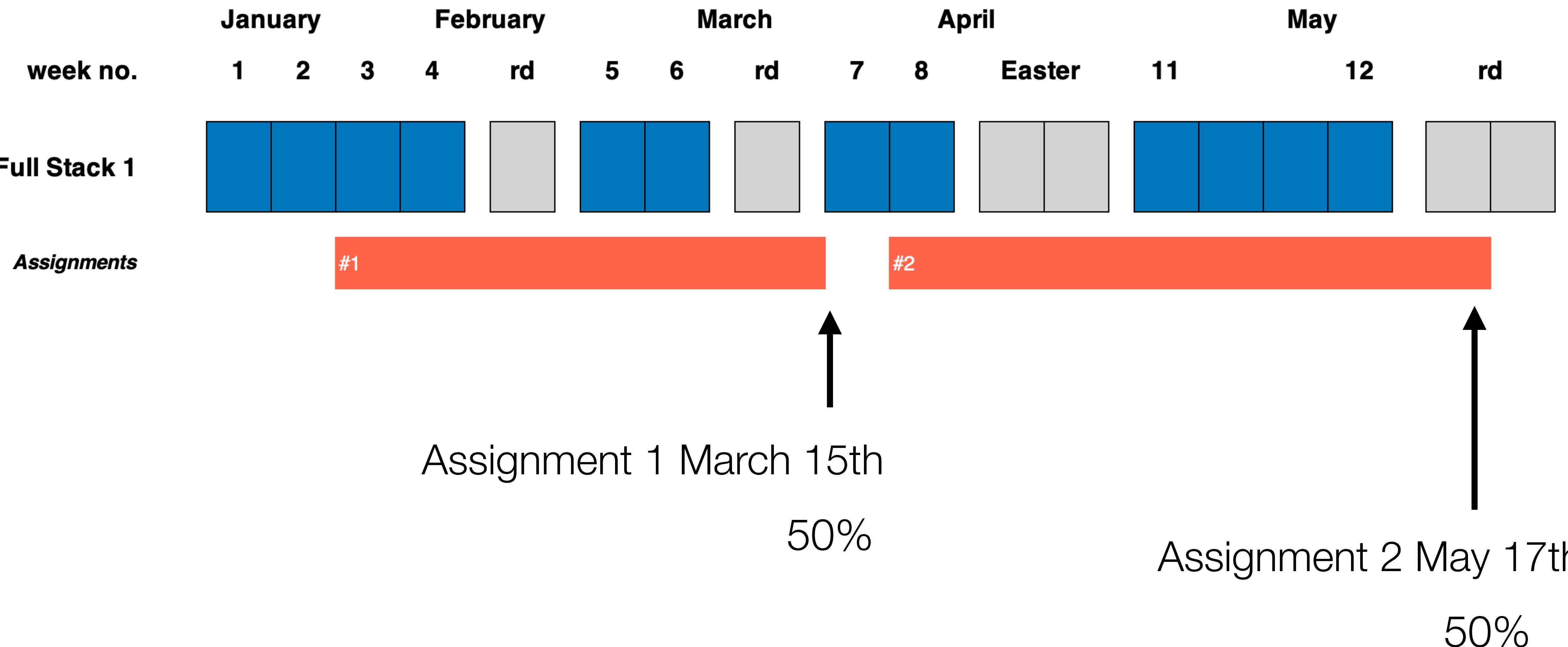


SSR patterns in SvelteKit

**24: Full Stack Development**



Patterns for Full Stack in SvelteKit



Lecture for the Topic  
Slides or External references/repos used in the lecture

Links to video chapter from main lecture

The screenshot shows a digital notebook interface with three main sections:

- Introducing Node**: A slide featuring logos of SE TU and OTH Regensburg, a photo of Eamonn de Leistar, and a diagram of a full stack developer role.
- Introducing Node & HAPI**: A slide with four cards: "Full Stack Web Development", "Node Context", "The Hapi Framework", and "Hapi Building Blocks".
- Introducing Playtime**: A slide with two cards: "Code Quality Tools" (listing ESLint, Airbnb, and Prettier) and "Model View Controller in HAPI" (explaining the MVC architecture pattern).

Annotations include a blue arrow pointing from the "Lecture for the Topic" text to the first slide, and a grey arrow pointing from the "Links to video chapter from main lecture" text to the "Introducing Playtime" section.

The screenshot shows a digital notebook interface with three cards:

- Learning Resources**: A card for a "Node Introduction" podcast generated by an AI LLM.
- Lab-01-Playtime-0-1-0**: A card for the first version of a lab implementing user signup and simple global playlists.
- LLM Notebook**: A card for a pretrained LLM on the topic.

AI Generated Summary Podcast (10 mins)

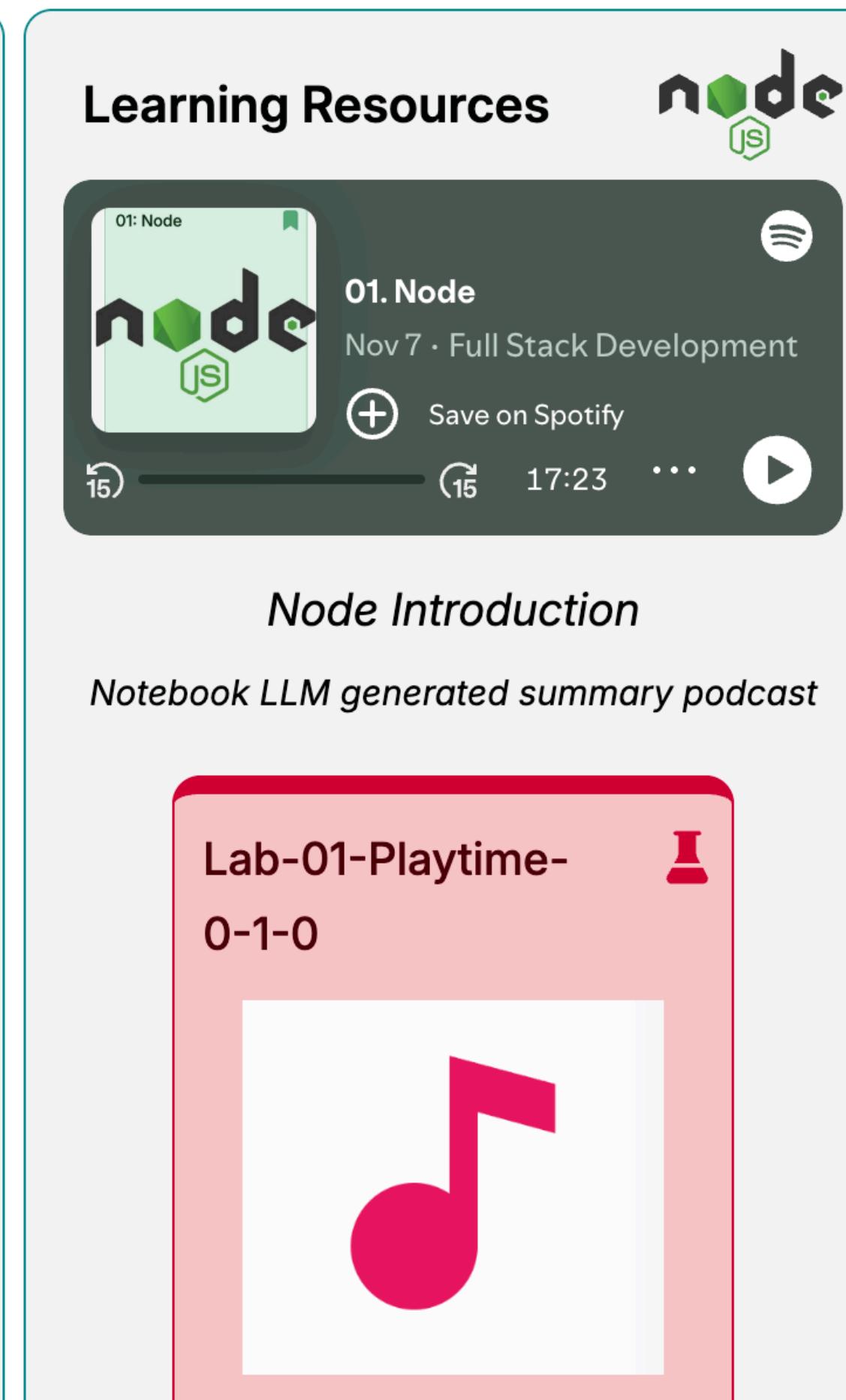
Lab for the Topic

Pre-trained LLM on this specific topic

Typical Topic Layout



vanced Full  
tack Web  
development



AI Generated Summary  
Podcast (10 mins)

Experimental

Perhaps listen as background  
like a podcast

Potentially useful overview/  
review of each topic

LLM Generated Podcast

notebooklm.google.com

01 Node Public

+ Create notebook Settings ⚙️ e

Sources Chat Studio

Select all sources ✓

00-overview.pdf ✓

01-node-essence.pdf ✓

02-hapi-philosophy.pdf ✓

03-hapi-building-blocks.pdf ✓

04-code-quality-tools.pdf ✓

05-playtime-0.1.0.pdf ✓

Node Development and the Hapi Fram... ✓

https://full-stack-oth-25.netlify.app/llm... ✓

# 01 Node

8 sources

This course material introduces **Advanced Full Stack Web Development** with a primary focus on the **Node.js** ecosystem and the **Hapi** framework. The sources outline essential concepts such as **asynchronous programming**, modular design, and the "frontend backend" architectural pattern. Students are guided through building a sample application called **Playtime**, which utilizes **Model-View-Controller (MVC)** patterns and in-memory data storage. To ensure high development standards, the curriculum integrates **code quality tools** like ESLint, Prettier, and the Airbnb style guide. Detailed laboratory instructions provide a practical roadmap for setting up a server, managing **web routes**, and implementing user authentication. Ultimately, the materials emphasize **convention over configuration** to help developers create scalable, maintainable, and secure web services.

How does Node's asynchronous model differ from traditional synchronous server-side programming in Java?

What are the core architectural building blocks and plugin philosophy of the Hapi framework?

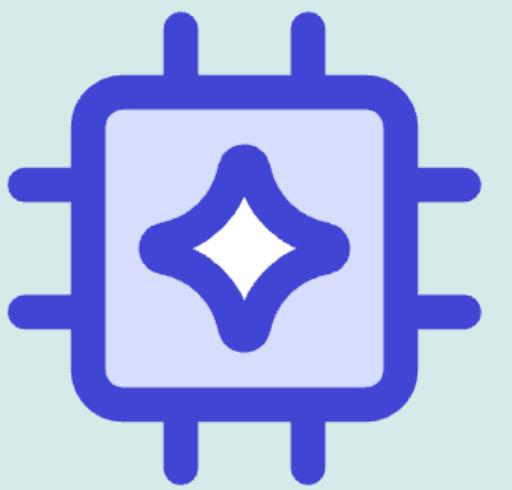
How is the Model-View-Controller pattern structured and implemented within a modular Node application?

Start typing... 8 sources ➔

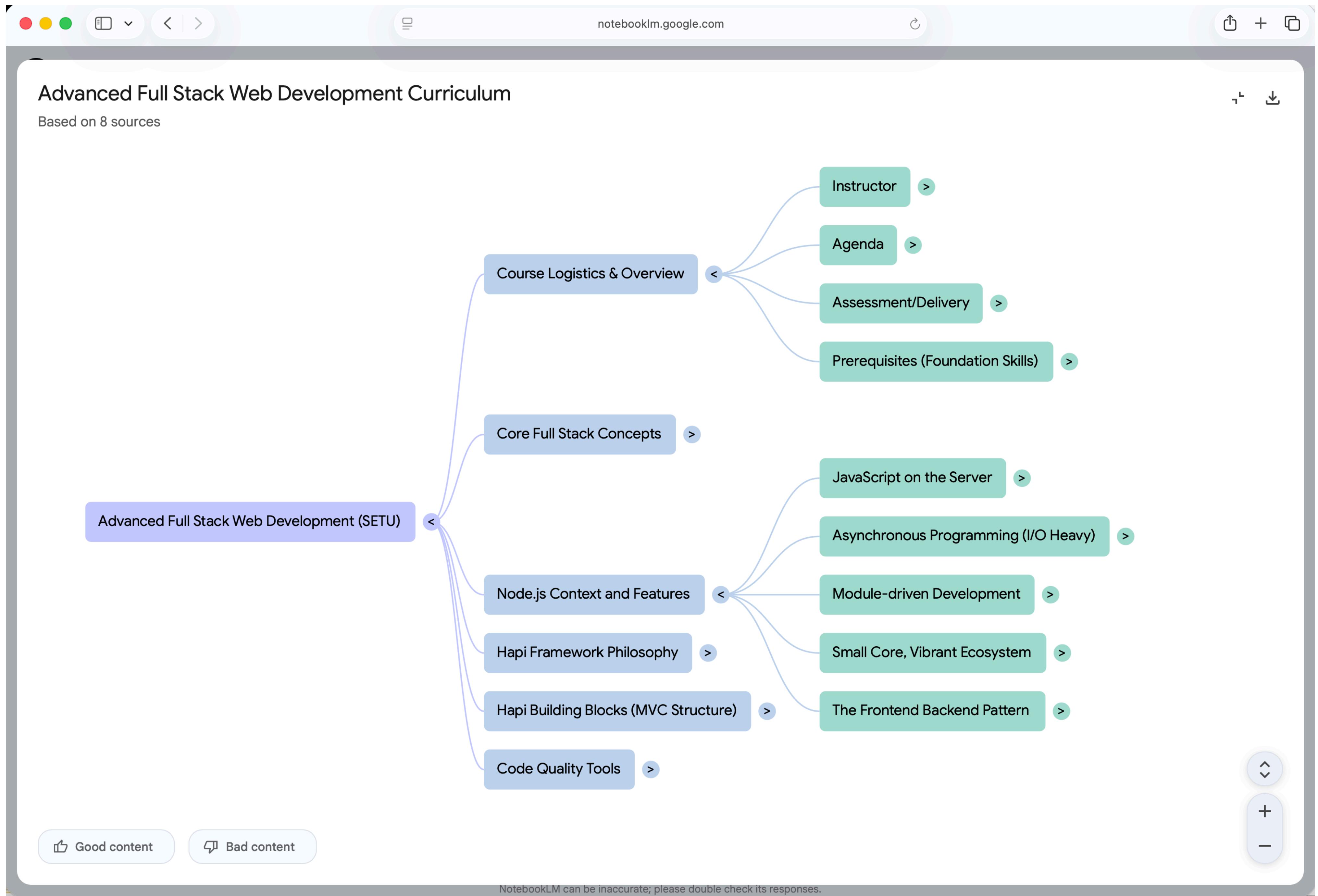
NotebookLM can be inaccurate; please double check its responses.

Pre-trained LLM on this specific topic

### LLM Notebook



A pretrained LLM on this topic



- 20 minutes max. Optional attendance.

- Very brief overview of Topic

- General Q&A

- Opportunity to check-in on module status

- “Hang back” time after 20 minutes for 1-to-1 consultation

- Live, NOT recorded

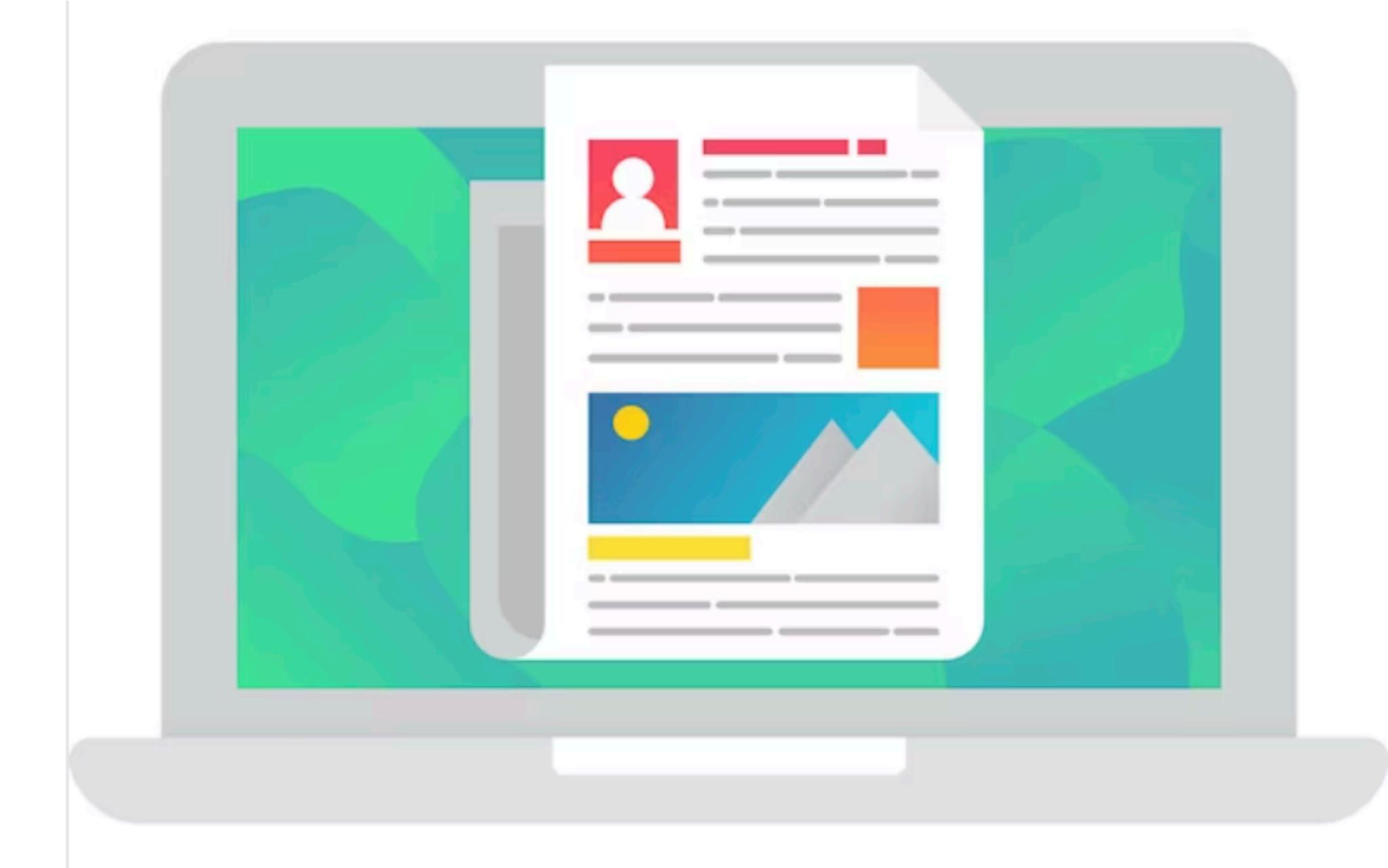
- Drop in as suits schedule

## Weekly Scheduled Zoom Sessions

MONDAY	TUESDAY	WEDNESDAY	FRIDAY
10:45			
12:30	<i>Full Stack 1</i> Topic Overview	<i>Full Stack 1</i> Topic Overview	
12:50			

# Assignment Motivation & Philosophy

- Assignments present multiple opportunities:
  - Crystallise knowledge on module
  - Opportunity for creativity & innovation
  - Potentially important portfolio contribution
- It is not over when submitted, presented and graded
  - Ongoing maintenance & enhancements
  - Periodically rebuild using new technology stack



The screenshot shows the Conduit application running in a web browser. The interface includes a navigation bar with 'Home', 'Sign in', and 'Sign up' links, and a main area titled 'conduit' with the subtitle 'A place to share your knowledge.' Below this, there's a 'Global feed' section showing posts from 'Chengqi' and 'Kontstantin'. To the right, there's a 'Popular tags' sidebar with various tags like 'react', 'node', 'angular', etc., each with a small icon and a count.

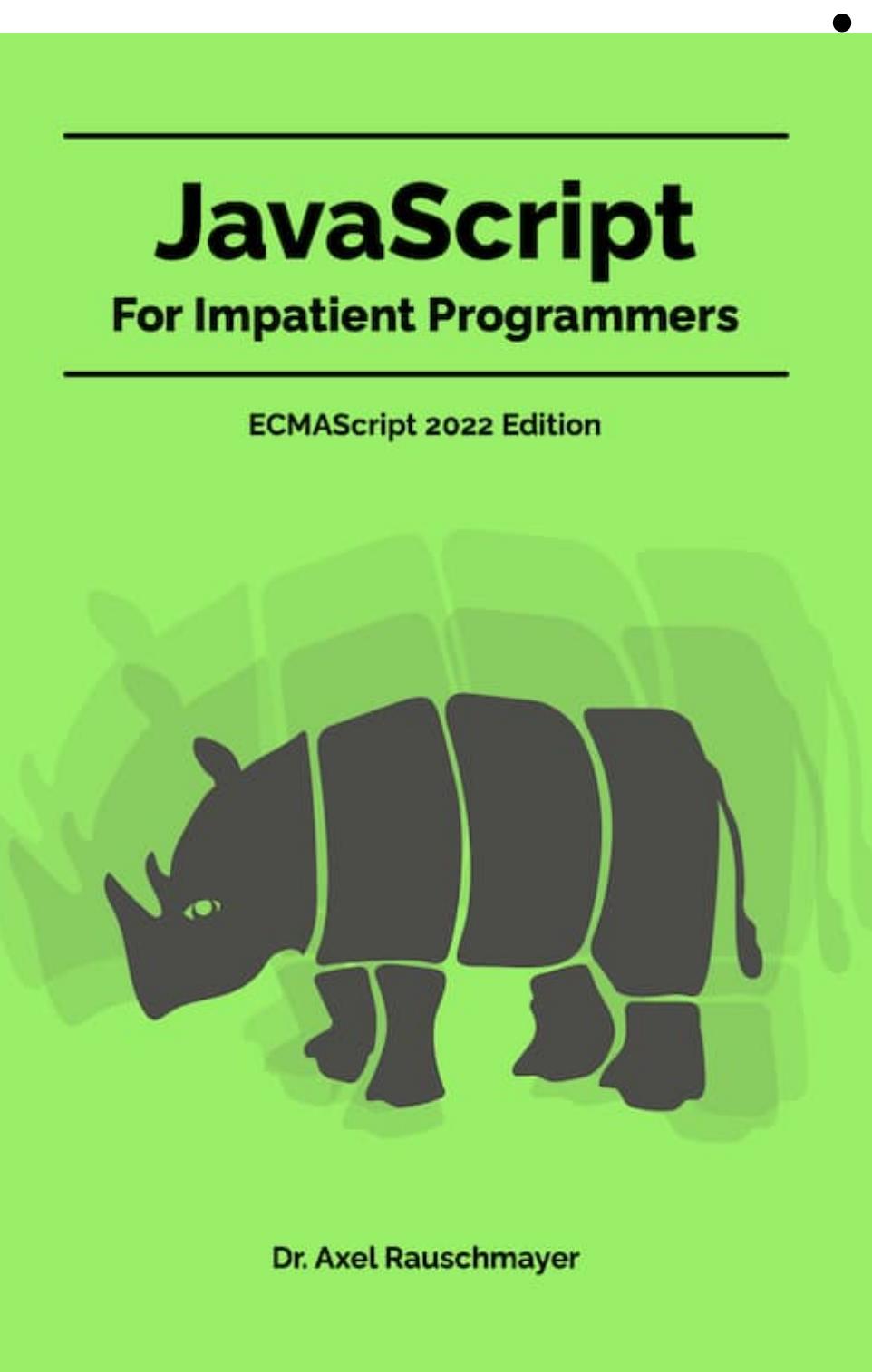
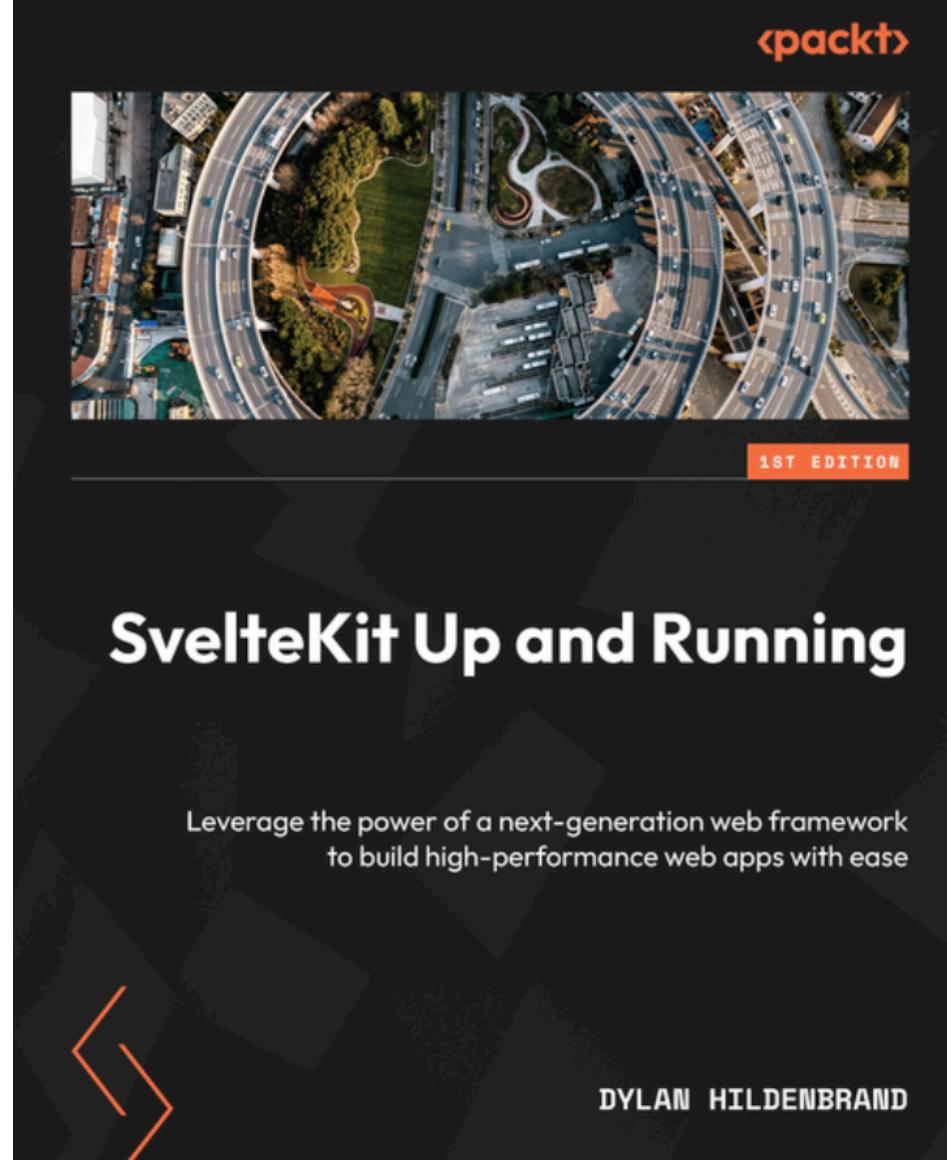
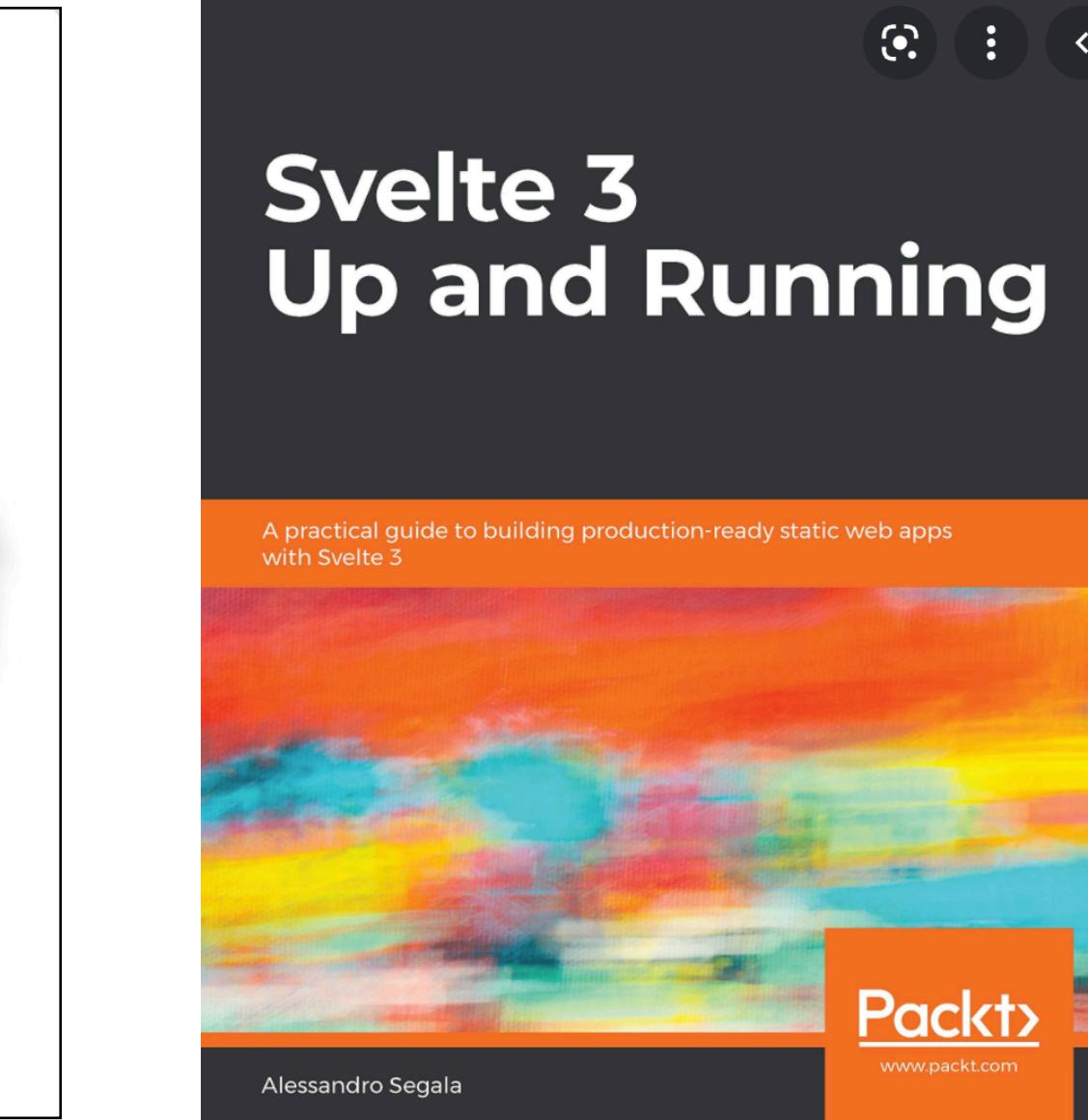
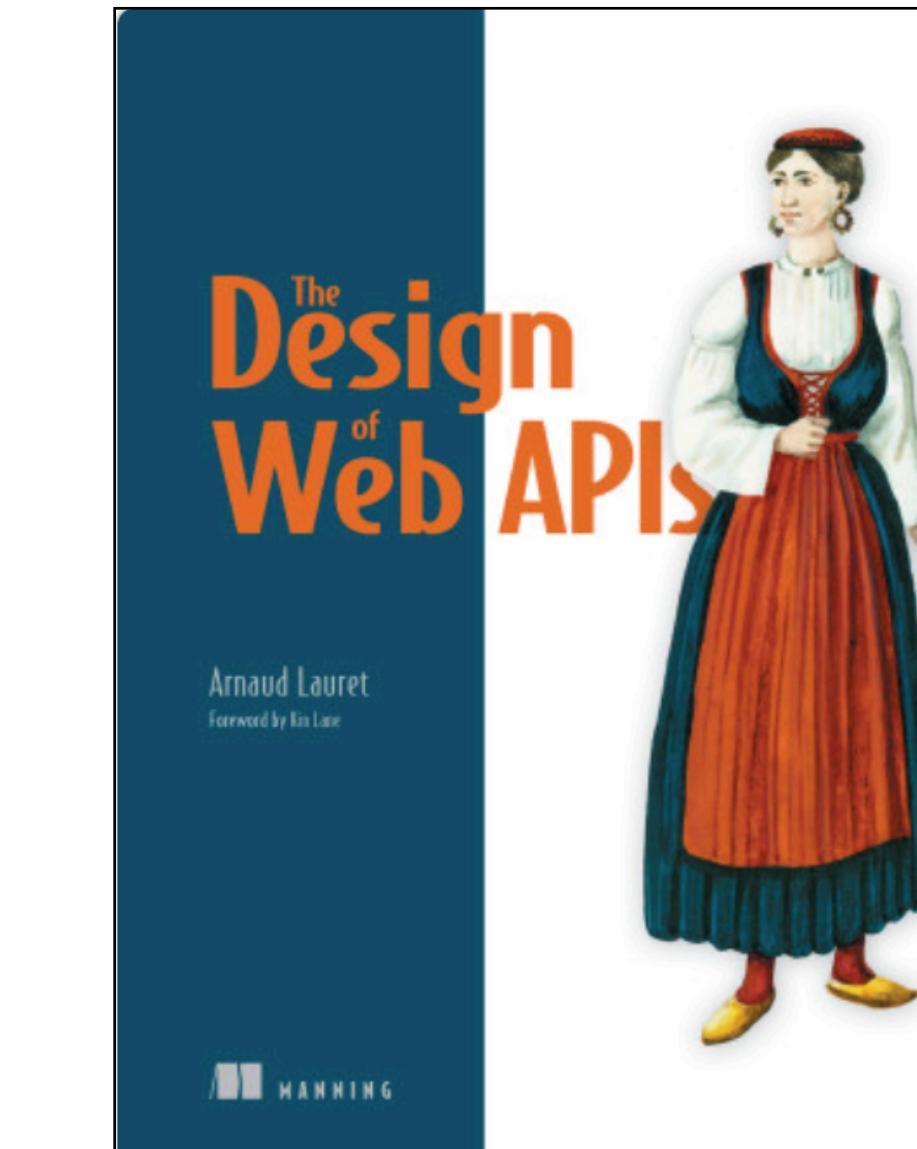
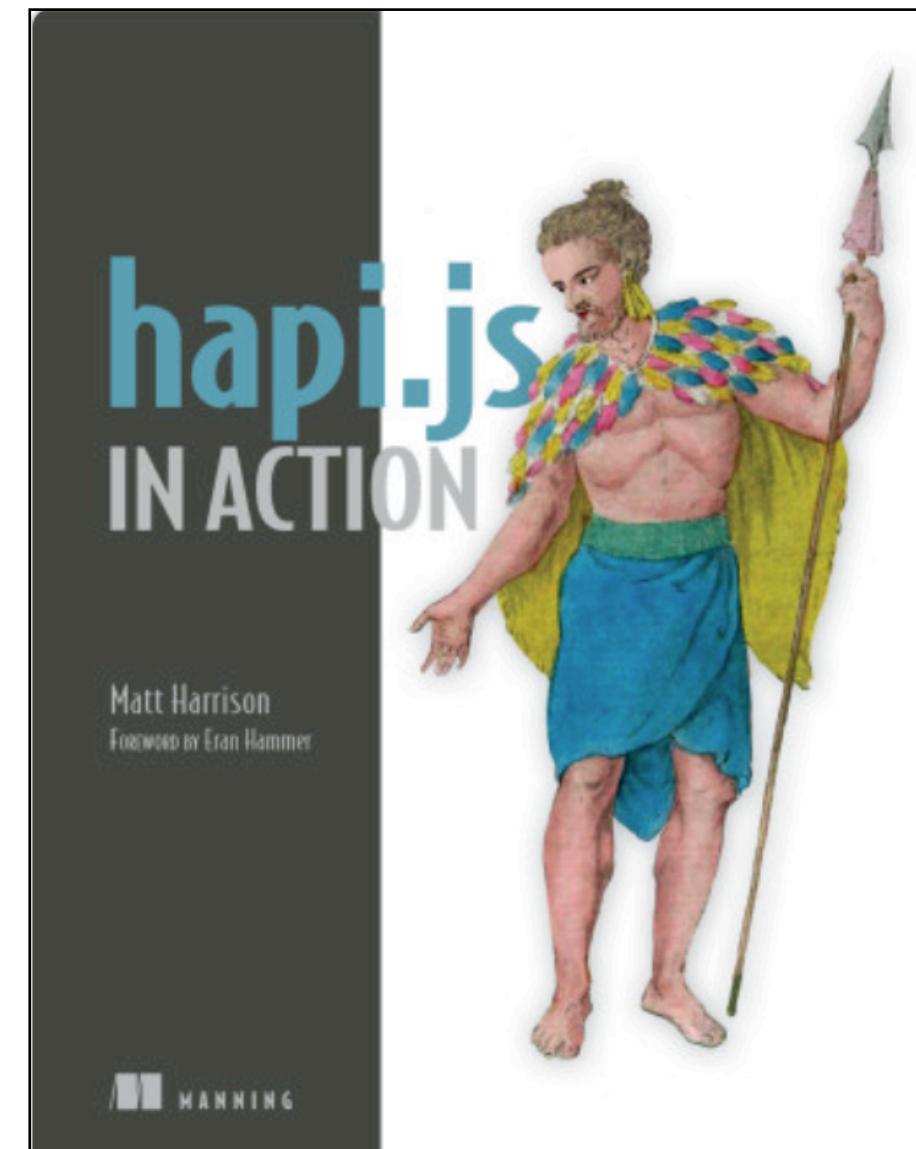
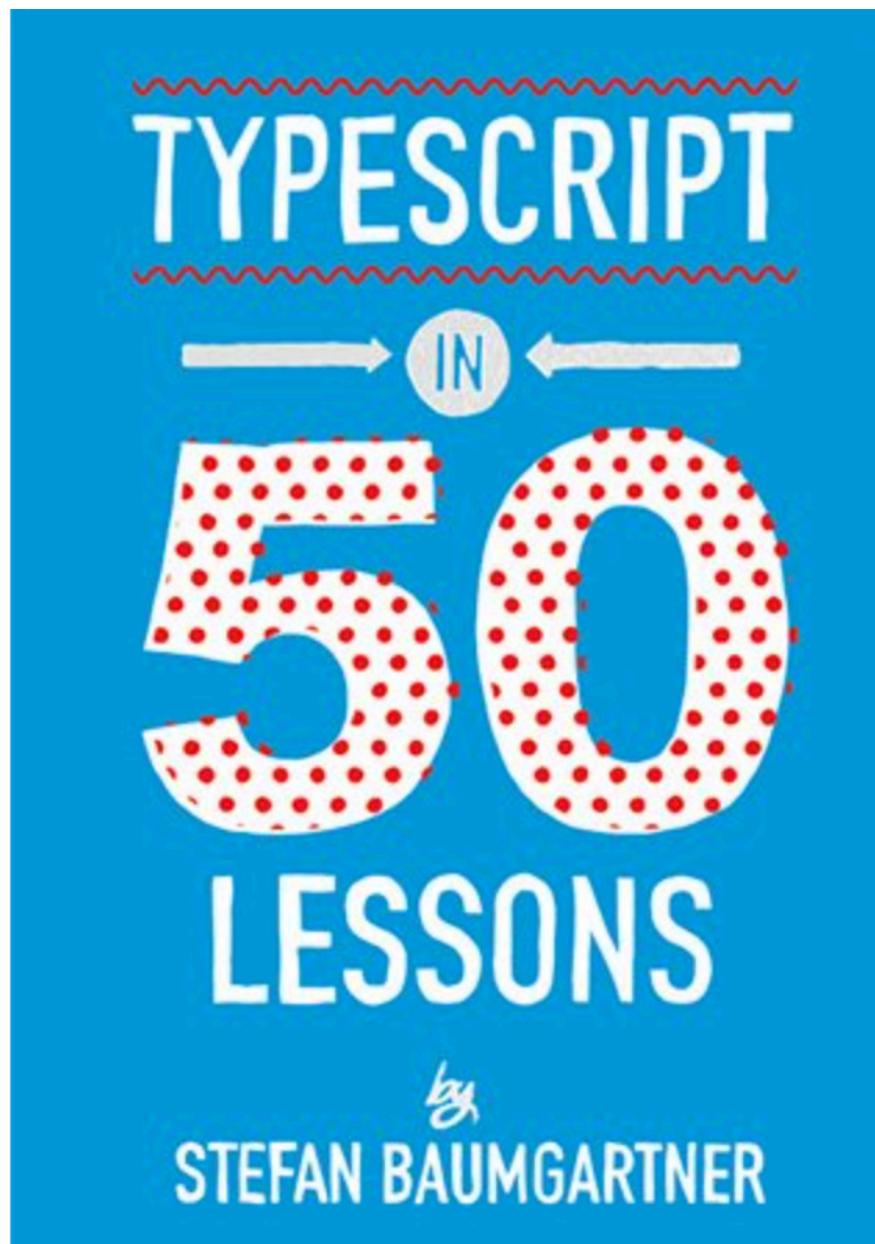
**The mother of all demo apps**

See how the exact same application is built using different libraries and frameworks.

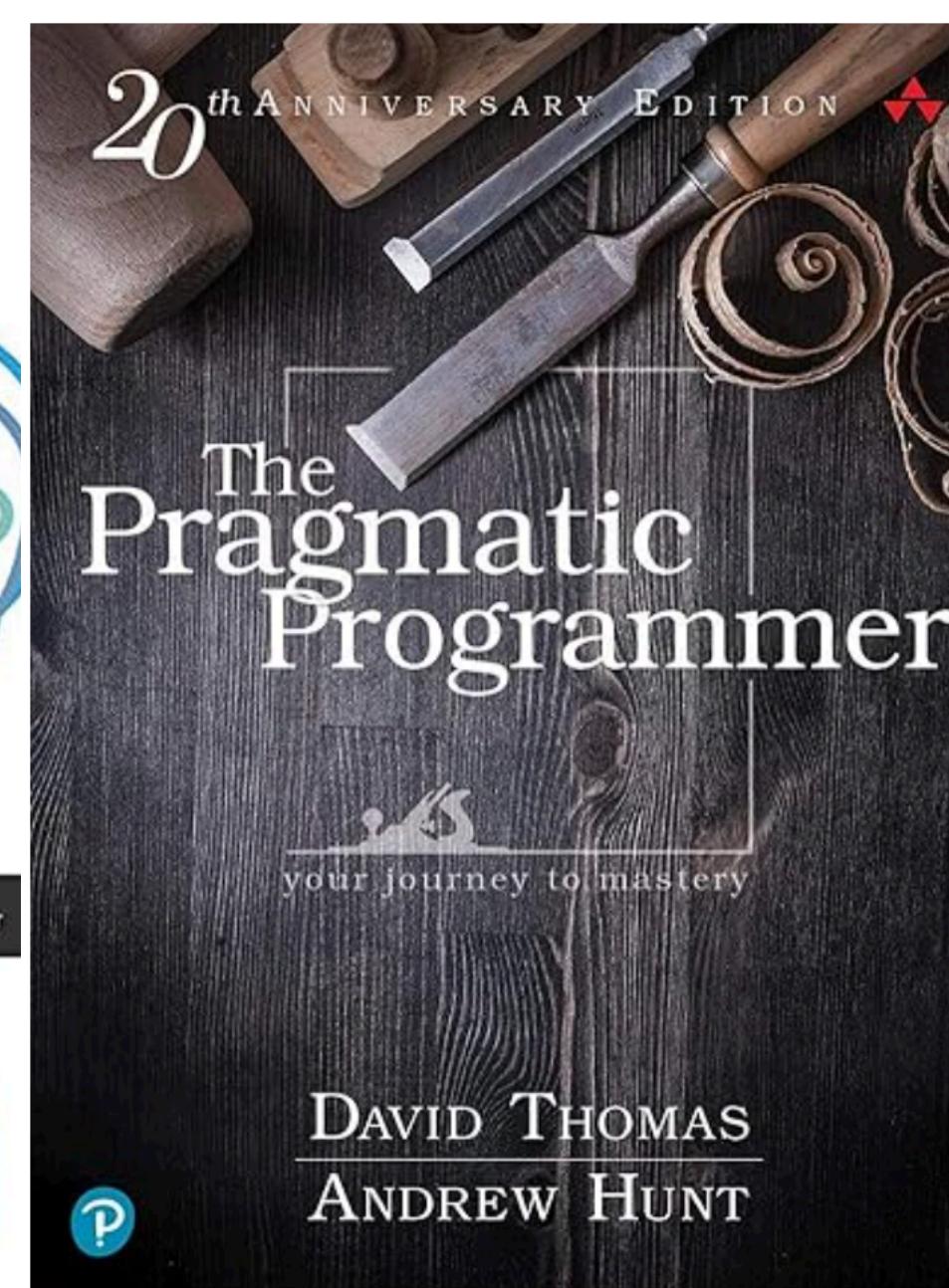
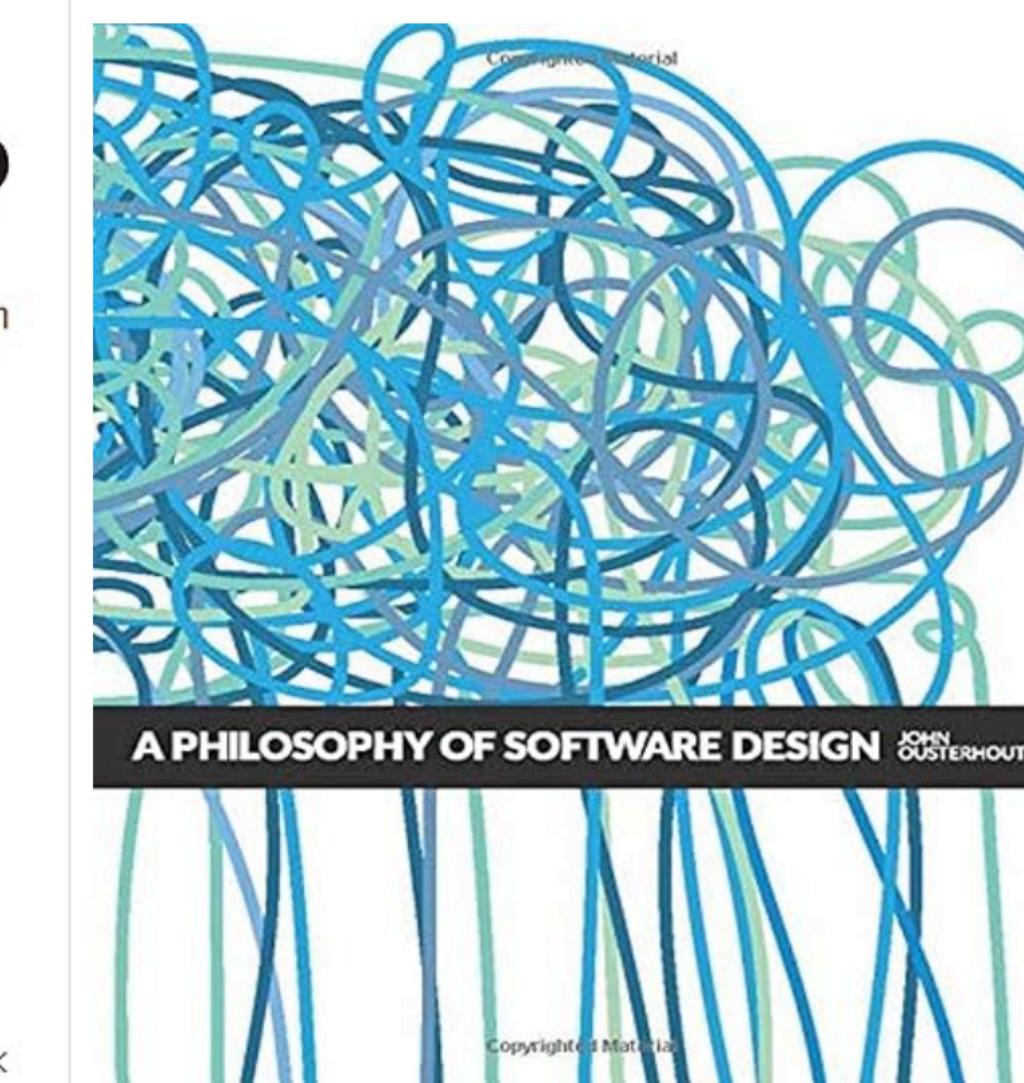
Frontend Backend Fullstack

<https://github.com/gothinkster/realworld>  
<https://codebase.show/projects/realworld>

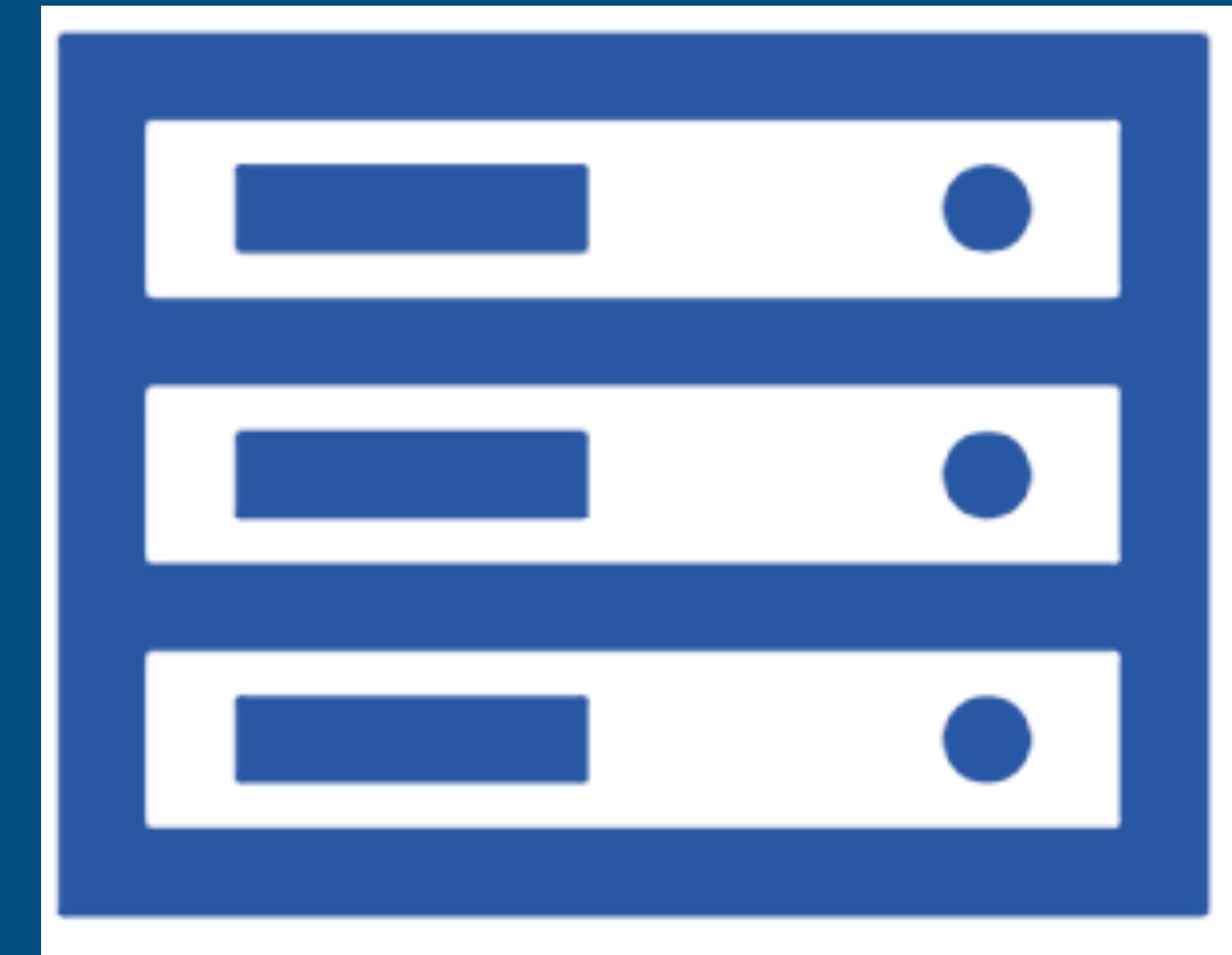




- <https://nodeweekly.com/>
- <https://javascriptweekly.com/>
- <https://frontendfoc.us/>
- <https://svelte.dev/blog>



# Course Overview



Full Stack Development 1