**Current Learnings Lesson Plans:**

Front-end ----

**Lesson 1**

Overview: Project Creation && Formatting

Library: Testing Library, Prettier

Code: None

Testing: Finding a element via text

Config: .vscode folder with extensions/settings, Prettier

CI/CD: basic yaml file, setup/checkout/run tests

Plugin: Material Icon Theme, Jest, & Prettier

Hook: none

**Lesson 2**

Overview: Consume Swagger API && Linter

Library: Swagger Typescript API

Code: Container Pattern

Testing: Update Tests for data fetch

Config: Git Attributes & ESLintRc, ESLintIgnore

CI/CD: add lint

Plugin: ESLint & ESLint type checking, Prettier ESLint

Hook: useDataFetch for API calls (use fetch API for now)

**Lesson 3**

Overview: Unit vs. Integration Testing

Library: Cypress

Code: TBD

Testing: Add integration tests (nav to page, confirm elements)

Config: Cypress.config & Cypress tsconfig

CI/CD: build app + add cypress

Plugin: Cypress ESLint + overrides

Hook: none

**Lesson 4**

Overview: Understand Storybook && Accessibility + Compatibility

Library: Storybook

Code: Button? TBD

Testing: Add tests to storybook

Plugin: Storybook ESLint, BrowsersListRc

CI/CD: build storybook

Config: Storybook config + Storybook plugins, BrowsersListRc, ESLint A11y, ESLint Compat

Hook: TBD

**Lesson 5**

Overview: Finish Configuration + Routing

Library: Husky, Axios, React Router

Testing: Test Coverage

Code: TBD + basic routing

Plugin: CSpell, Editorconfig

CI/CD: deploy app + storybook

Config: editorConfig, husky, pretty-quick, cspell, update swagger for axios

Hook: TBD

Back-end ---

**Lesson 1**

Overview: Project Creation && See a working API

Library: Swagger

Code: API Specific (annotations/attributes for Swagger)

Testing: Simple Unit Test for Actions/Controllers

Pattern: Facade

Config: (if .NET, editorConfig)

CI/CD: setup/checkout

**Lesson 2**

Overview: Services & DI

Library: (Mockito, Moq)

Code: (What to Register as What, (Singleton, Transient, Scoped))

Testing: Introduce Mocking

Pattern: Proxy

Config: TBD

CI/CD: restore/build/run tests

**Lesson 3**

Overview: Data Access, Repo, layed architecture

Library: (MediatR, Java TBD)

Code: Repository Pattern, Folders/Naming/Namespaces

Pattern: Repository, Mediator

Testing: Update for code

CI/CD: publish, package

**Lesson 4**

Overview: Domain in respects to Sprockets

Library: Fluent Validation

Code: Domain Objects, Validation

Patterns: Entity, Value Object

Testing: Update for domain stuff

CI/CD: deploy

**Lesson 5**

Overview: Database TBD?

Library: Hiberate, EntityFramework, ADO.NET

Code: TBD

Patterns:

Testing: Integration Testing

CI/CD: Integration Testing in pipeline