Contents

[🧠 Case Study: Learning Tracker Web App 2](#_Toc195446052)

[1. Problem Statement 2](#_Toc195446053)

[2. Target User 2](#_Toc195446054)

[3. User Stories 2](#_Toc195446055)

[4. Core Features 2](#_Toc195446056)

[🎨 Design Phase 3](#_Toc195446057)

[1. Entities / Data Model 3](#_Toc195446058)

[2. Pages / Routes 3](#_Toc195446059)

[3. UI Wireframes (Text Format) 3](#_Toc195446060)

[Dashboard Page 3](#_Toc195446061)

[Entry Detail Page 4](#_Toc195446062)

[Technical Design 4](#_Toc195446063)

[Backend Stack (suggestion) 4](#_Toc195446064)

[Frontend Stack (suggestion) 4](#_Toc195446065)

[Optional Enhancements 4](#_Toc195446066)

That sounds like a great project! Let’s break it down into a **case study** and then move into a **design** phase with both UI/UX and system architecture considerations.

# 🧠 Case Study: Learning Tracker Web App

## 1. Problem Statement

As a developer or learner, it's easy to lose track of online tutorials, platforms (like YouTube, Udemy, or freeCodeCamp), and your current progress. You want a centralized system where you can:

* Keep a list of what you’re learning
* Track progress by episode, module, or completion
* Organize by platform and programming language
* Add notes, links, and filter efficiently
* Learning Path (adding sets of tutorials and provides a learning priority )

## 2. Target User

* Self-taught developers
* Students in bootcamps
* Anyone following online learning paths or tutorials

## 3. User Stories

**🧑‍💻 As a user, I want to:**

* Add a new tutorial I'm following
* Update my progress (e.g., "last watched episode 5")
* See all tutorials filtered by language/platform
* Visit a detailed view to see all info + notes
* Delete completed or dropped tutorials

## 4. Core Features

| **Feature** | **Description** |
| --- | --- |
| Add/Edit/Delete entry | CRUD for learning entries |
| Detail view per entry | Shows title, platform, programming language(s), link, notes, progress |
| Filtering/Searching | Filter by platform or programming language |
| Categorization | (Optional) Add categories or tags like "Beginner", "In-depth" |
| Responsive UI | Works on mobile and desktop |

## 5. Optional Features

| **Feature** | **Description** |
| --- | --- |
| Learning Path (If I have time) | Provides path so user knows sequence in which he needs to learn topics |
| Sorting |  |
| Filtering |  |
| User Authentication |  |
| Dashboard |  |
|  |  |

# 🎨 Design Phase

## 1. Entities / Data Model

We can design the following core models:

## 2. Pages / Routes

| **Page** | **Description** |
| --- | --- |
| / | Dashboard with list of all learning entries |
| /entry/:id | Detail view of a specific learning entry |
| /add or modal | Form to add a new learning entry |
| /edit/:id or modal | Edit form for an entry |
| /filter | UI filter or dropdowns for language/platform |

## 3. UI Wireframes (Text Format)

### Dashboard Page

+------------------------------------------+

| Filter: [Platform ▼] [Language ▼] |

+------------------------------------------+

| 📘 React for Beginners (YouTube) |

| ➤ Language: JavaScript, React |

| ➤ Last seen: Episode 5 |

| [View Details] [Edit] [Delete] |

+------------------------------------------+

| 📘 Python Web Scraping (Udemy) |

| ➤ Language: Python |

| ➤ Last seen: Lecture 12 |

| [View Details] [Edit] [Delete] |

+------------------------------------------+

### Entry Detail Page

Title: React for Beginners

Platform: YouTube

Languages: JavaScript, React

Category: Frontend

Link: https://youtube.com/react-course

Last Seen: Episode 5

Notes:

"This course covers modern hooks and good practices."

[Edit] [Back to Dashboard]

# Technical Design

## Backend Stack (suggestion)

* **Next.js**
* **mySQL**
* **Prisma**
* **Next Auth.js**

## Frontend Stack (suggestion)

* **React**
* **TailwindCSS**
* **Daisy**

# Optional Enhancements

* Add reminder to revisit a course if not updated in X days
* Sync progress across devices via login
* Markdown notes support
* Dark mode

# Development Road Map

## Core Development

* CRUD a course
* Detail view per entry
* Filtering/Searching
* Categorization
* Responsive UI

## Advance Development

* Learning Path (If I have time)
* Sorting
* Filtering
* User Authentication
* Dashboard