

What Is Remix?

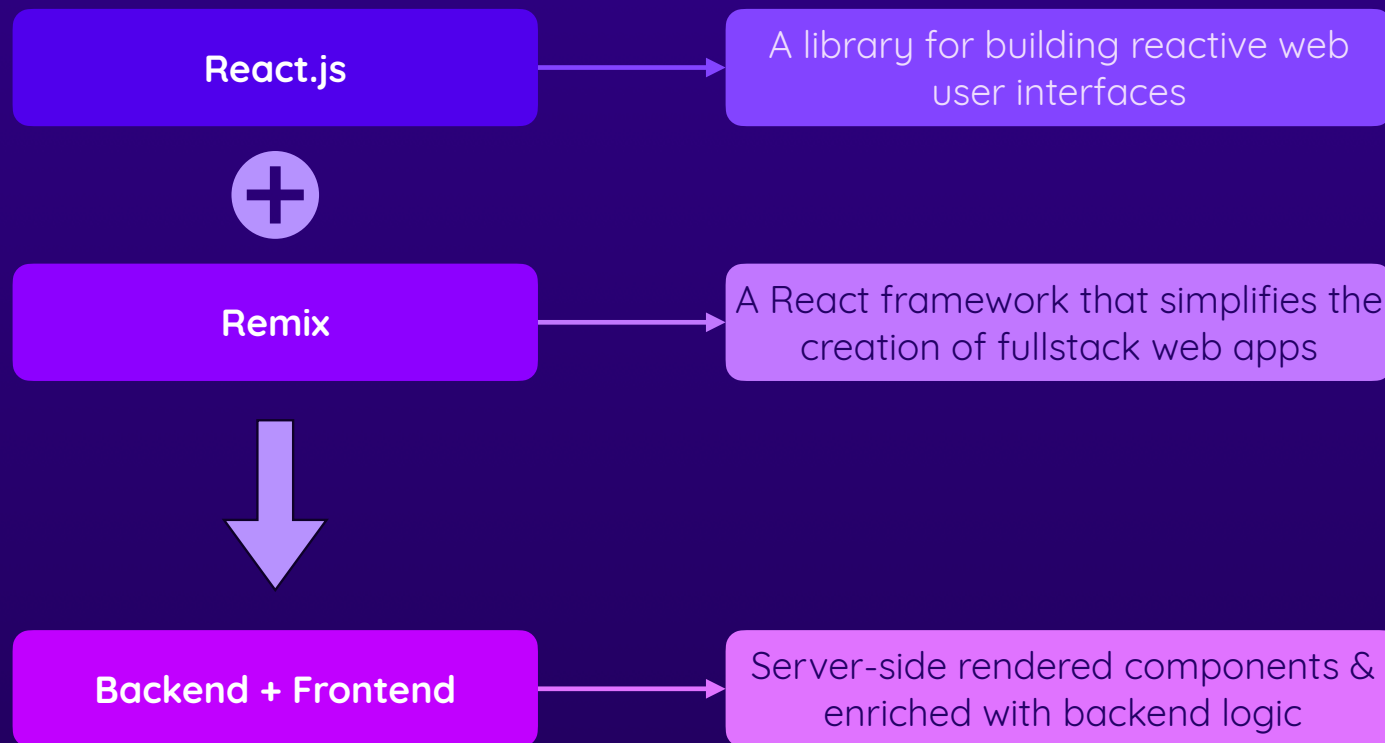
What Is Remix?

A React Framework
for building fullstack apps

Isn't React Already A Library?

Why Remix?

What Is Remix?



Remix vs “Just React”

Remix

Easily merge backend +
frontend logic

SEO & no project switching

Deployment: Server-side
code execution must be
supported

“Just React”

Just the frontend, separate
backend required

Separate projects, SEO is
tricky with just React

Deployment: Static hosting
(for just HTML, CSS + client-
side JS) suffices

Remix vs NextJS

Remix

Always server-side rendered

No static site generation
(build-time pre-rendering)

Always requires host that
supports server-side code
execution

NextJS

Optional server-side
rendering supported

Static site generation (at
build time) supported

Deployment options: Static
hosting vs server-side code
execution



About This Course



Essentials

The key Remix features
you must know

~2h



Deep Dives

Build up on basics + dive
deeper

~6h

About This Course



Routing & Page
Rendering



Data Fetching



Data Mutation

CRUD & Error Handling



Layouts & Styling



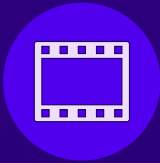
Authentication



SEO & Deployment



How To Get The Most Out Of This Course



Watch the Videos

At your pace: Use the video player controls

On-Demand: Repeat videos & sections as needed



Code Along & Practice

Pause & try things on your own

Practice what you learned (also in your own projects)

Use attached slides & code snapshots



Help Each Other

Ask & answer in the Q&A section

Join our amazing Discord community!

React Essentials

Crucial Fundamentals

- ▶ What & Why?
- ▶ Working with Components & Data
- ▶ State & Effects

What Is React?

Build Reactive User Interfaces



Declarative
(not imperative)

Define **intended results**, not
the steps that lead there



Component-focused

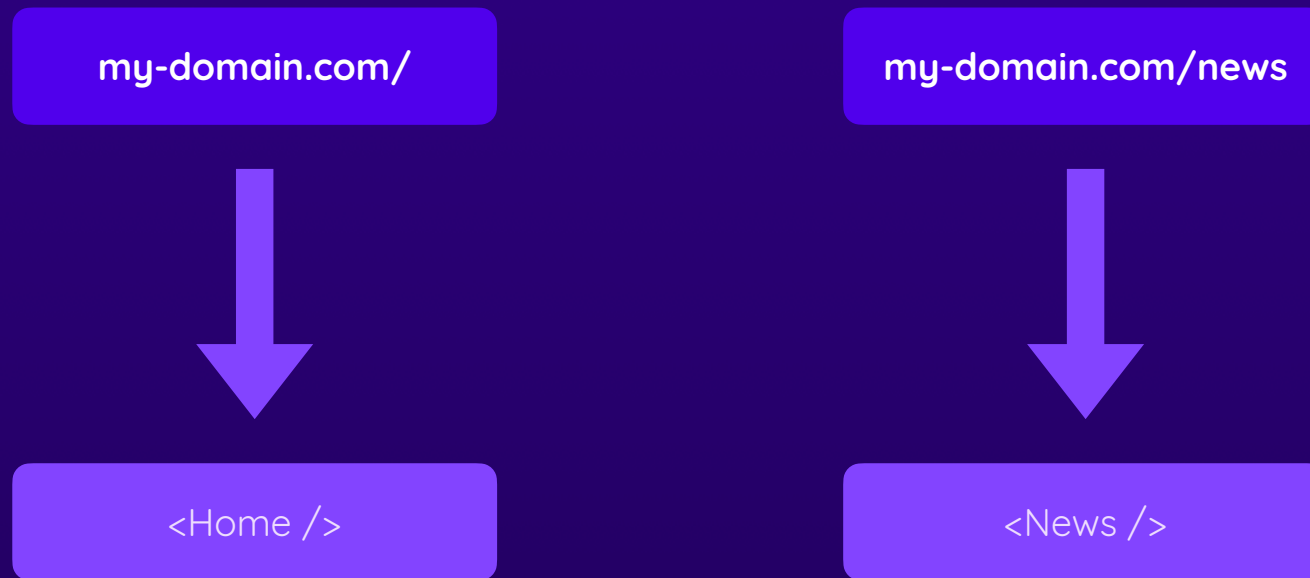
Compose the UI from many
small, **reusable building blocks**



Automatic UI updates
when needed

Define **different possible scenarios** and let React update
the UI automatically

Building Multi-Page Apps (Routing)



Remix Essentials

Crucial Fundamentals

- ▶ Project Structure & Routing
- ▶ Data Fetching & Manipulation
- ▶ Styling & Metadata

Routing & Layouts: Deep Dive

Always Serving The Right Pages

- ▶ Defining Routes & Layouts
- ▶ Different Kinds of Routes
- ▶ Nested Routes & Non-Page Routes



Project Work

1

Route Setup

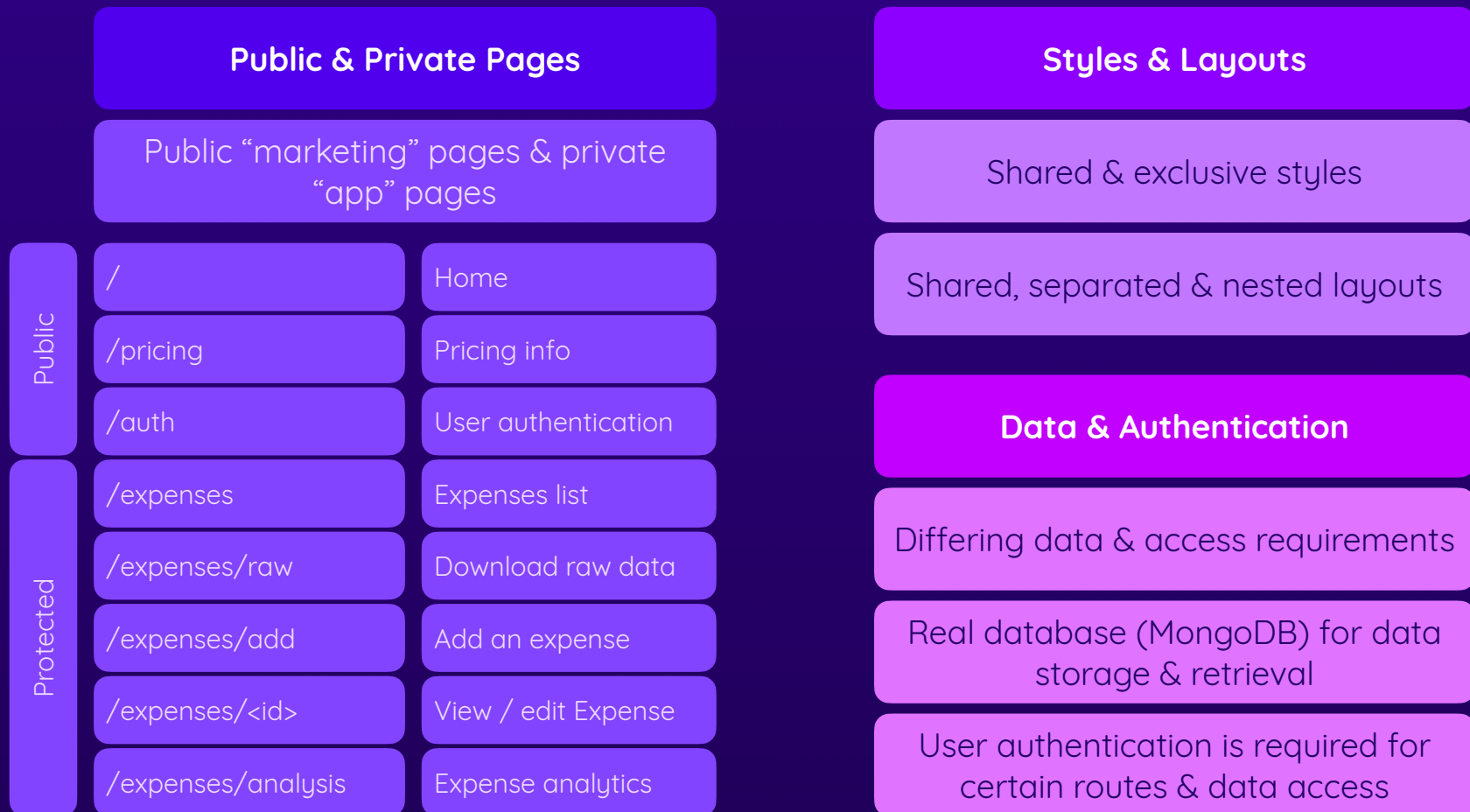
2

Adding Non-Page Components

3

Layouts & Styles

The Course Project



Remix Routing

routes/index.jsx



/

routes/pricing.jsx



/pricing

Layout Routes

Routes that are active whilst
other routes are active as well



Route components that include
other route components

Root route: Always active

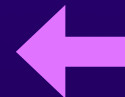
root.jsx

+

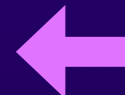
/routes/auth.jsx

+

/routes/expenses.jsx



/auth



/expenses

Root Isn't The Only Layout Route!

You can add as many layout routes
("route wrappers") as needed



`<roulename>.jsx + /<roulename>`

`expenses.jsx + /expenses`

/expenses Layout Route

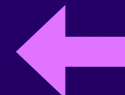
`expenses.jsx`

+

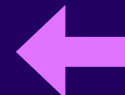
`expenses/$id.jsx`

+

`expenses/index.jsx`



`/expenses/expense-1`



`/expenses`

Navigation Paths

Absolute Path

/expenses

Gets appended right after the domain



<your-domain>/expenses

Relative Path

expenses

Gets appended after the currently active route path



<your-domain>/list/expenses

(if navigation was initiated on
<your-domain>/list)

Supported Route Types

Basic Routes	→	news.jsx	/news
Nested Routes with Folders	→	news/create.jsx	/news/create
Nested Routes with Dot Delimiters	→	news.create.jsx	/news/create
Dynamic Routes	→	/news/\$id.jsx OR /news.\$id.jsx	/news/abc
Splat Routes	→	/news/\$	/news/match/any/path
Layout Routes	→	/news/create.jsx + news.jsx	/news/create (with shared layout)
Pathless Routes	→	/__news/create.jsx + __news.jsx	/create (with shared layout)

Data Fetching & Mutations

Working with (Backend) Data

- ▶ More on “loader” & “action”
- ▶ Data Submission: Alternatives to <Form>
- ▶ Sharing Data Between Routes

Authentication

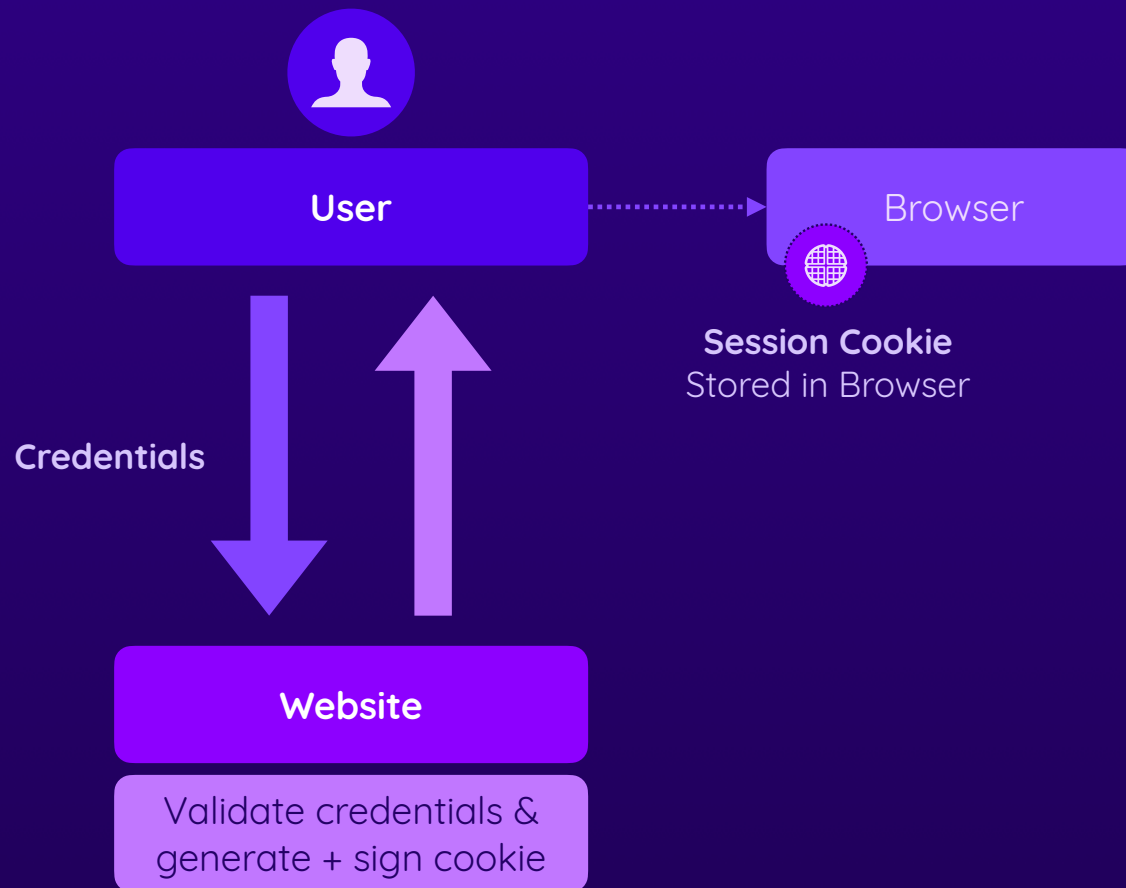
Working with Users

- ▶ Adding Signup & Login
- ▶ Managing Authentication Sessions
- ▶ Protecting Pages & Data

How Does User Authentication Work?



How Does User Authentication Work?



How Does User Authentication Work?

