ACADEMIND

What Is Remix?

ACADEMIND

What Is Remix?

A React Framework

for building fullstack apps

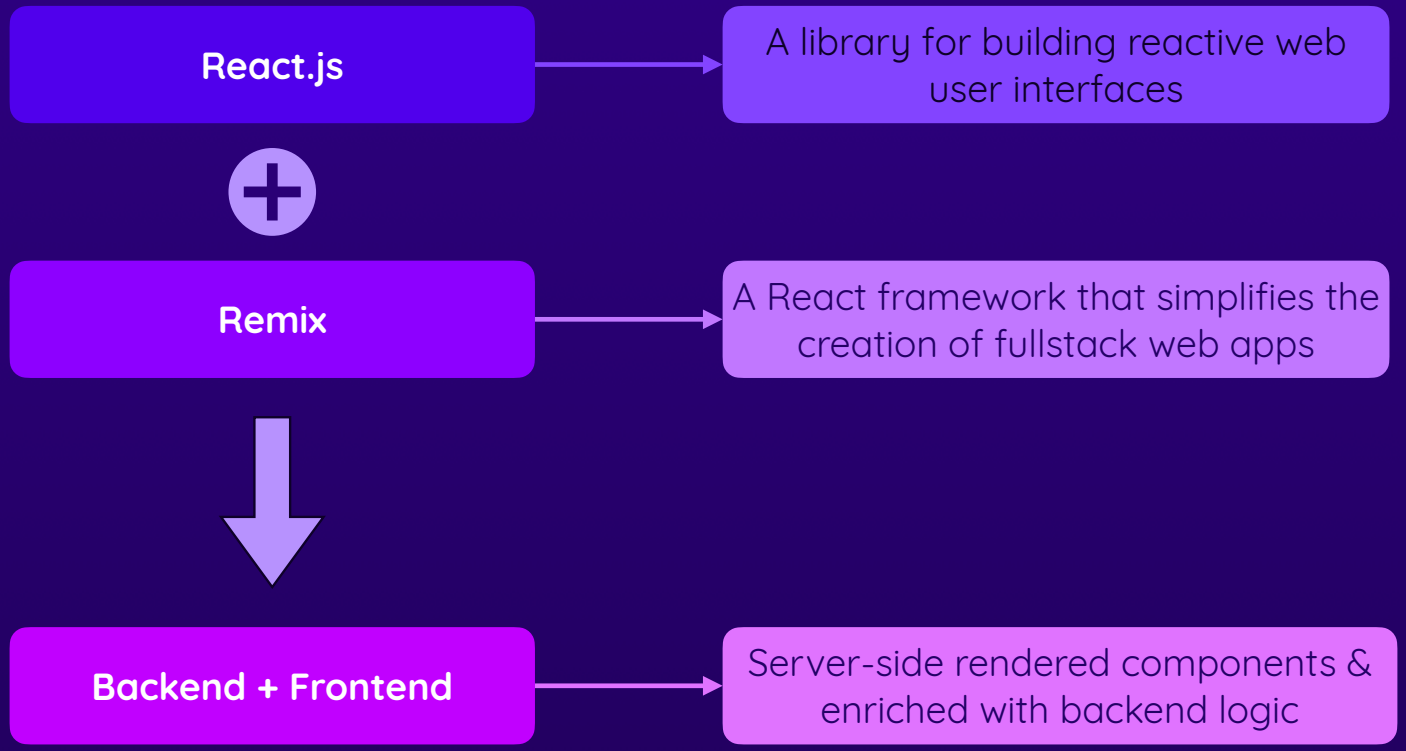
ACADEMIND

Isn't React Already A Library?

ACADEMIND

Why Remix?

What Is Remix?



ACADEMIND

Remix vs “Just React”

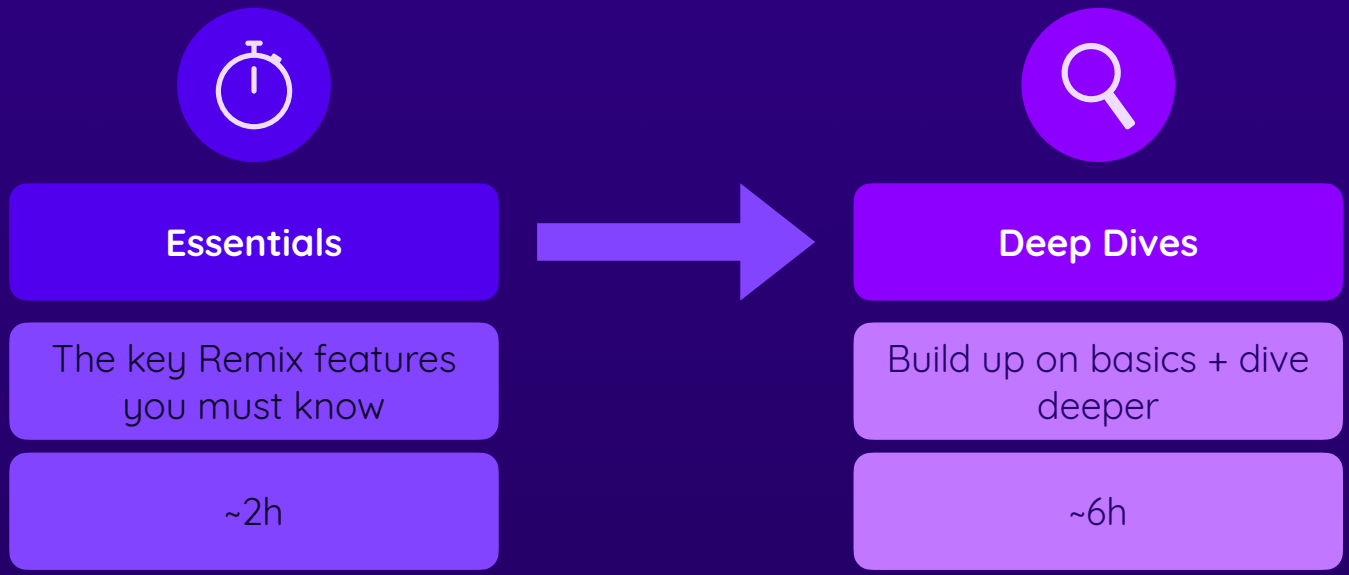
Remix	“Just React”
Easily merge backend + frontend logic	Just the frontend, separate backend required
SEO & no project switching	Separate projects, SEO is tricky with just React
Deployment: Server-side code execution must be supported	Deployment: Static hosting (for just HTML, CSS + client-side JS) suffices

ACADEMIND

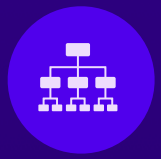
Remix vs NextJS

Remix	NextJS
Always server-side rendered	Optional server-side rendering supported
No static site generation (build-time pre-rendering)	Static site generation (at build time) supported
Always requires host that supports server-side code execution	Deployment options: Static hosting vs server-side code execution

About This Course



About This Course



Routing & Page Rendering

Data Fetching

Data Mutation

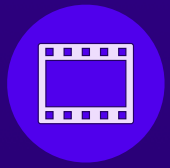
CRUD & Error Handling

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!

ACADEMIND

React Essentials

Crucial Fundamentals

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

ACADEMIND

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)

ACADEMIND

Remix Essentials

Crucial Fundamentals

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

ACADEMIND

Routing & Layouts: Deep Dive

Always Serving The Right Pages

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

ACADEMIND

Project Work

1. Route Setup
2. Adding Non-Page Components
3. Layouts & Styles

The Course Project

Public & Private Pages

Public “marketing” pages & private “app” pages

Public	/	Home
Public	/pricing	Pricing info
Public	/auth	User authentication
Protected	/expenses	Expenses list
Protected	/expenses/raw	Download raw data
Protected	/expenses/add	Add an expense
Protected	/expenses/<id>	View / edit Expense
Protected	/expenses/analysis	Expense analytics

Styles & Layouts

Shared & exclusive styles

Shared, separated & nested layouts

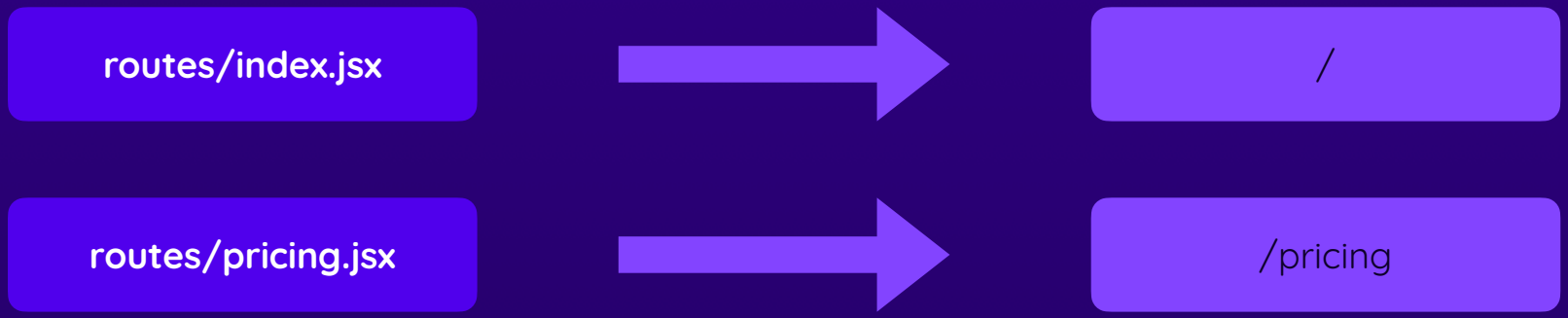
Data & Authentication

Differing data & access requirements

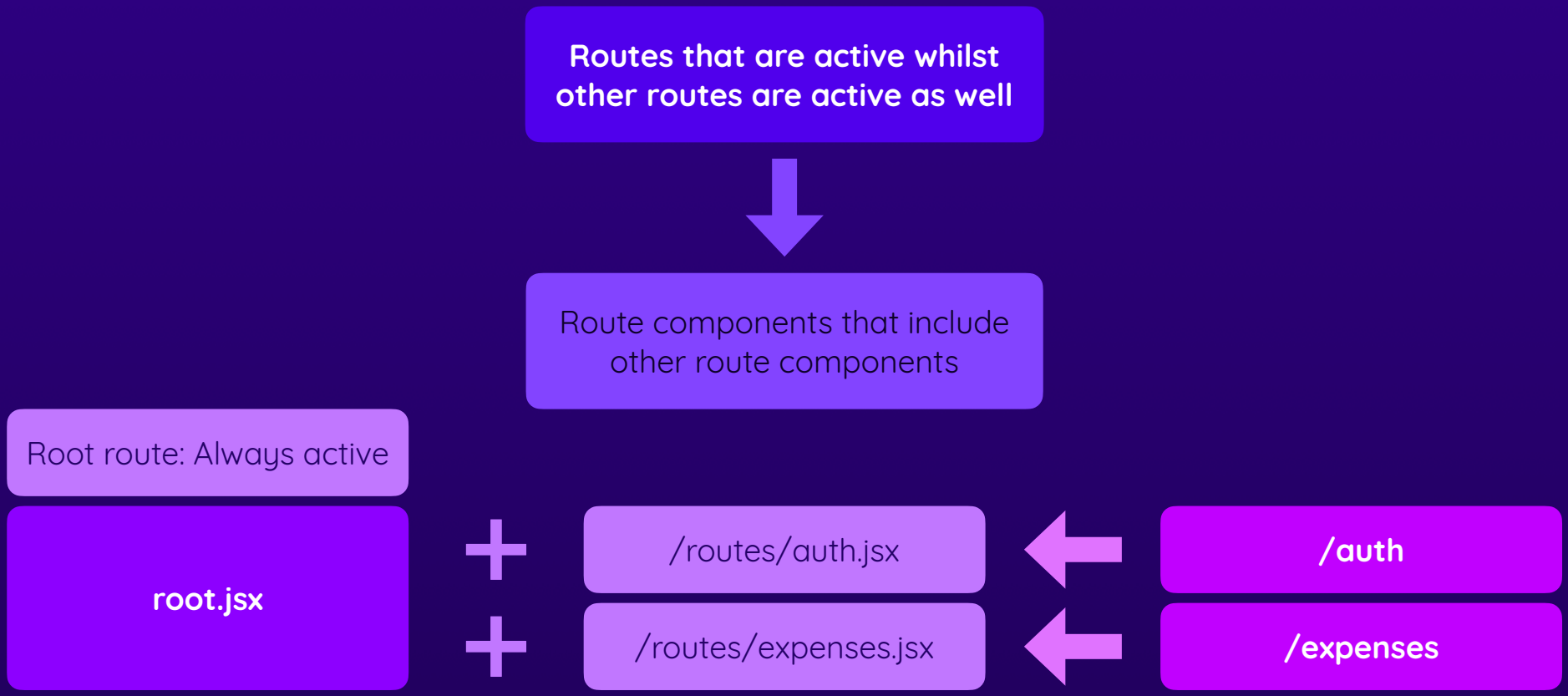
Real database (MongoDB) for data storage & retrieval

User authentication is required for certain routes & data access

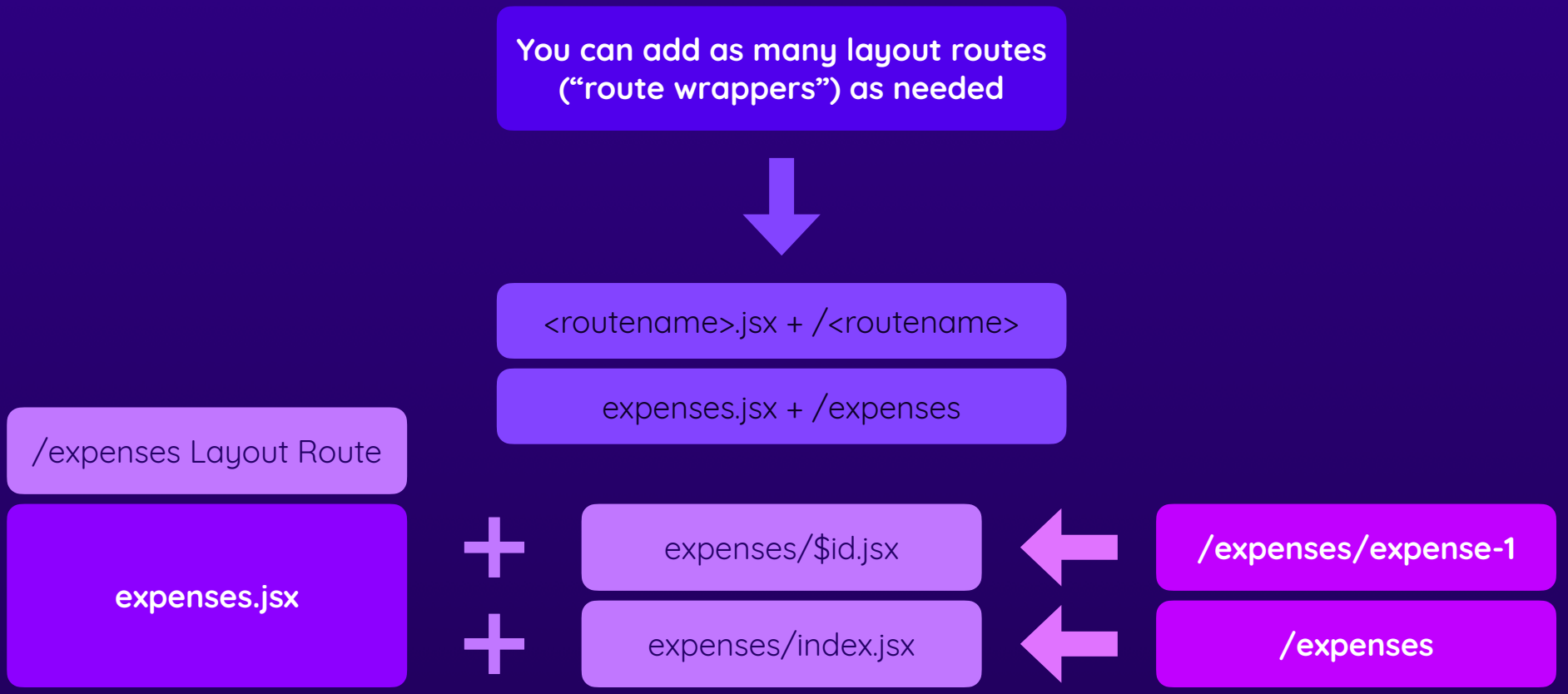
Remix Routing



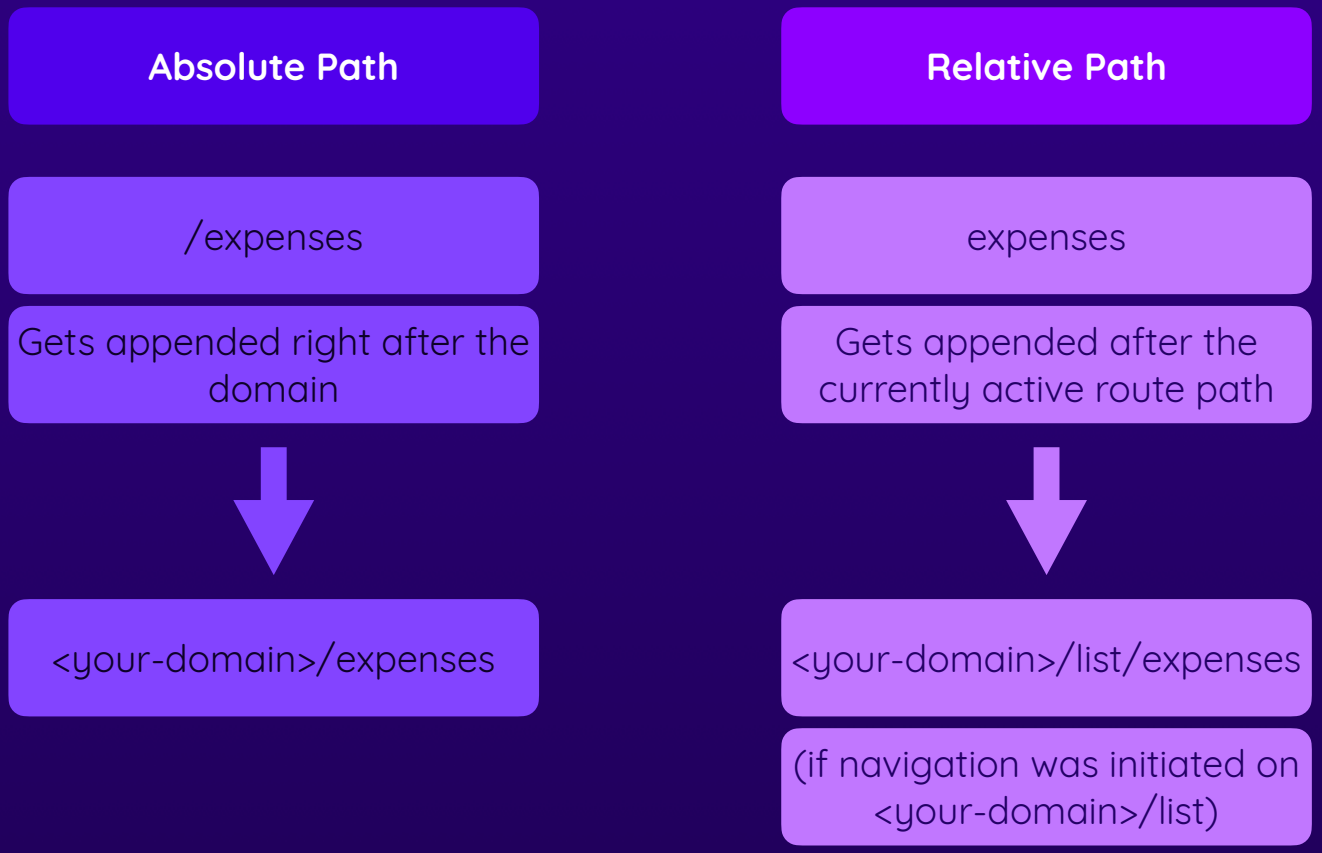
Layout Routes



Root Isn't The Only Layout Route!



Navigation Paths



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)

ACADEMIND

Data Fetching & Mutations

Working with (Backend) Data

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

ACADEMIND

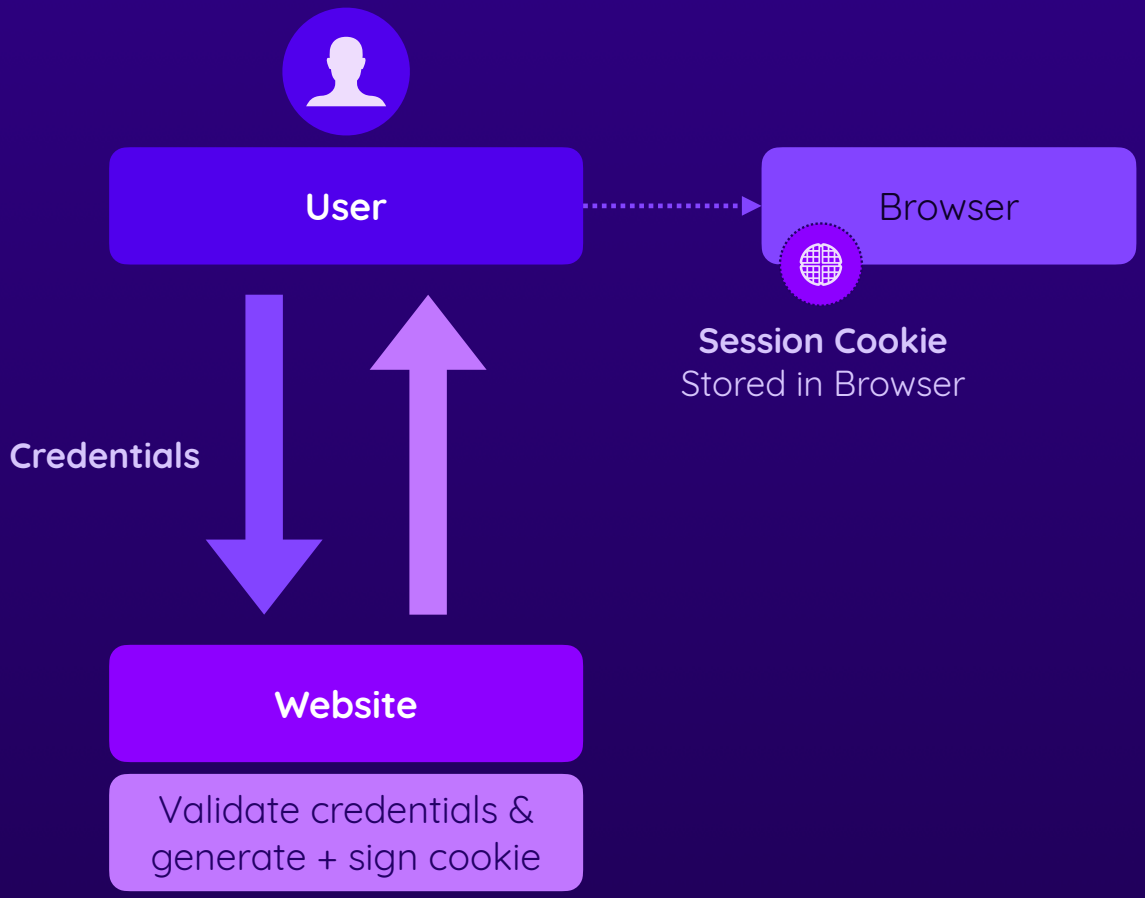
Authentication

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?