6. JavaScript Application Development

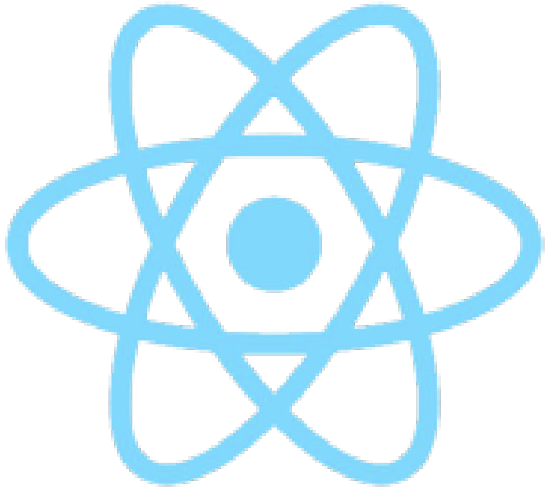
Adrian Adiaconitei

LINK Academy

Obiective

- ✓ Recapitulare React
  - ✓ Ce este React?
  - ✓ Ce este JSX?
  - ✓ Ce este o componentă?
  - ✓ Props și State
  - ✓ React Hooks
- ✓ ReactCRUD App
  - ✓ Navigarea între pagini
  - ✓ Formulare
  - ✓ Comunicarea cu API-uri

React



React

React

- ✓ **React** este o biblioteca JavaScript pentru construirea de interfete.(**FrontEnd**)
- ✓ **React Native** este un întreg cadru pentru construirea de aplicații multiplatforme, fie că este web, iOS sau Android (cross-platform mobile framework)
- ✓ Este creat și întreținut de Facebook
- ✓ React.JS - V18.2.0 (2022).

React - JSX (JS XML)

- ✓ JSX este un limbaj de templating inventat de echipa care a facut React pentru a putea să îmbine JavaScript cu HTML/XML (**Syntax Extension to JavaScript**.)

https://legacy.reactjs.org/docs/introducing-jsx.html

React

- ✓ În React, template-urile se numesc elemente
- ✓ Elementele sunt compuse din componente
- ✓ Randarea unui element se face cu metoda **render**

React - Componente

- ✓ Componentele vă permit să vă împărțiți interfața în “piese” reutilizabile,
- ✓ Conceptual, componentele sunt asemanatoare cu funcțiile/clasele JavaScript. Acestea acceptă intrări/proprietăți (denumite “**props**”) și returnează elemente React, care descriu ceea ce ar trebui să apară în interfața.

React - Componente

Functional Components

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Class Components

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

React - State

- ✓ Stare (State) reprezintă un obiect JavaScript care păstrează informația influentând aspectul sau starea componentei compilate și randate. Diferența față de **props** este aceea că starea este menținută în contextul componentei - similar cu definirea unei variabile în interioriul unei funcții.
- ✓ În cazul componentelor funcționale, o variabila de stare este declarată cu ajutorul useState (React Hooks).

LINKAcademy

React

SN	Props	State
1.	Props are read-only.	State changes can be asynchronous.
2.	Props are immutable.	State is mutable.
3.	Props allow you to pass data from one component to other components as an argument.	State holds information about the components.
4.	Props can be accessed by the child component.	State cannot be accessed by child components.
5.	Props are used to communicate between components.	States can be used for rendering dynamic changes with the component.
6.	Stateless component can have Props.	Stateless components cannot have State.
7.	Props make components reusable.	State cannot make components reusable.
8.	Props are external and controlled by whatever renders the component.	The State is internal and controlled by the React Component itself.

React Hooks

- ✓ Hook - disponibil din React 16.8.
- ✓ Hook - pot fi apelate numai în interiorul componentelor de tip funcție.
- ✓ Hook - pot fi apelate doar la nivelul superior al unei componente.
- ✓ Hook - nu vor funcționa în componentele de tip class.
- ✓ Hook - nu pot fi condiționate

React Hooks

- ✓ Un Hook este o funcție specială care vă permite să vă „conectați” la funcționalitățile oferite de React.

- ✓ **useState**
- ✓ **useEffect**
- ✓ **useContext**
- ✓ **useRef**
- ✓ **useReducer**
- ✓ **useCallback**
- ✓ **useMemo**

React Hooks

- ✓ useState este un Hook care vă permite să adaugați stare pentru componentele funcționale.
- ✓ Folosind useEffect Hook, ii spui Reactului că componenta ta trebuie să facă ceva după randare / la actualizarea unei variabile.

```
useEffect(() => { //Runs on every render });
useEffect(() => { //Runs only on the first render }, []);
useEffect(() => { //Runs on the first render //And any time any dependency value changes }, [prop, state]);
```

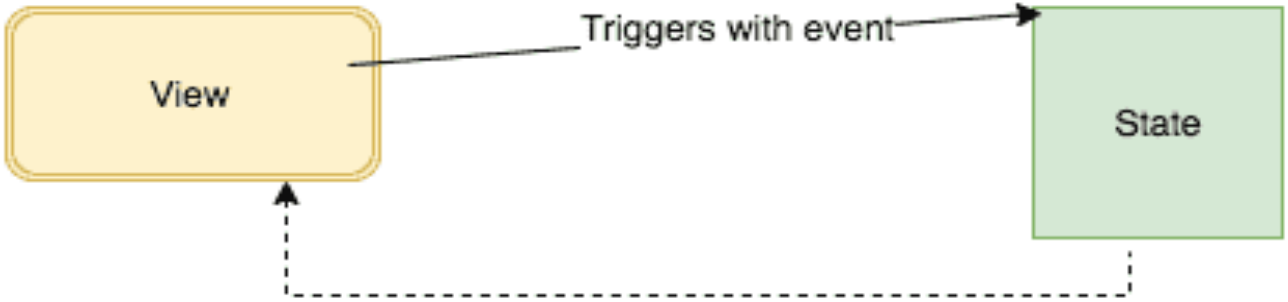
Demoapp => pages=> Blog.js

- ✓ useContext - este o modalitate de a gestiona state la nivel global. Transmiterea informației de la componenta părinte la componenta copil

Demoapp=>pages=> Autor.js

React

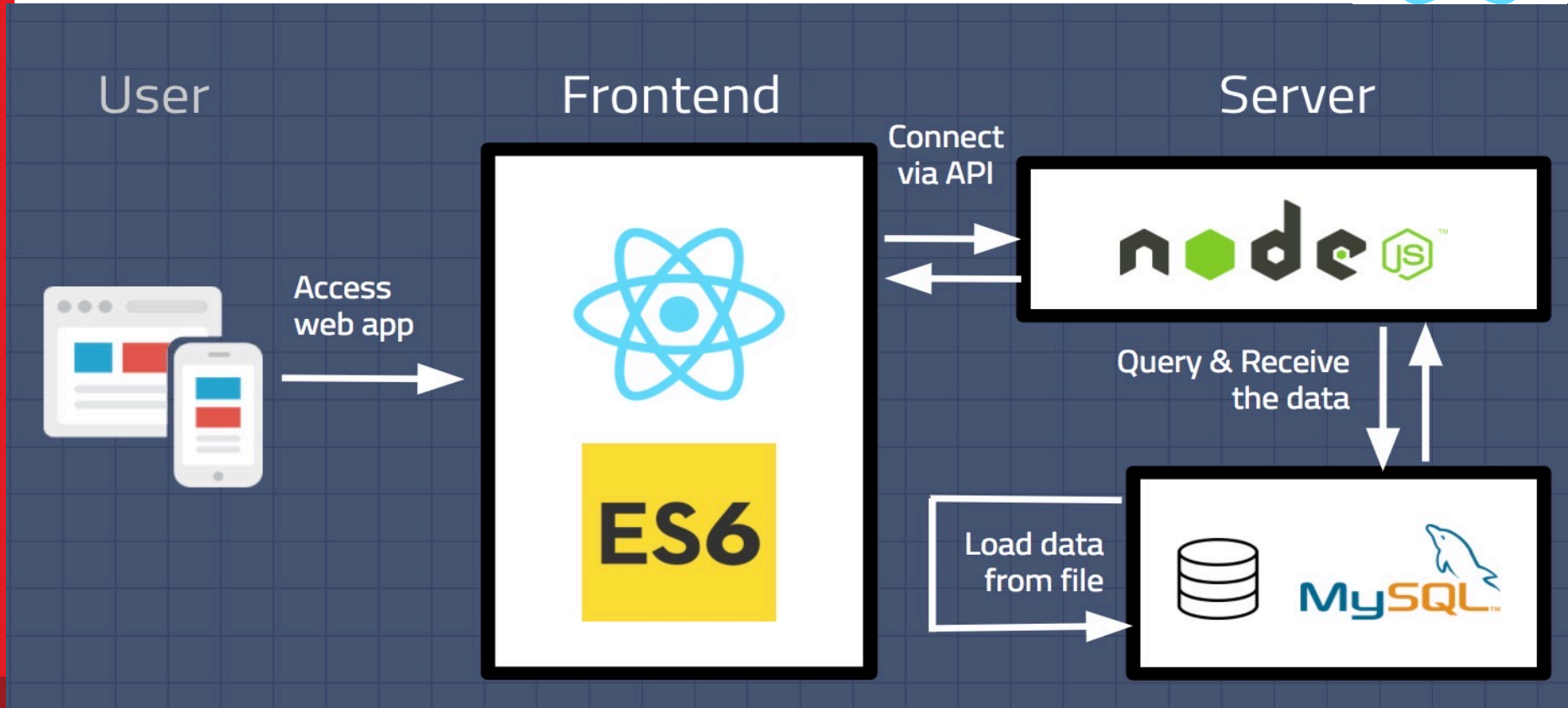
- ✓ One-way Data Binding:
  - ✓ Component => View: props și state
  - ✓ View => Component: event



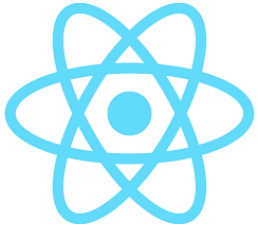
Exemplu:demoapp SalutComponent

LINKAcademy

ReactCRUD



ReactCRUD

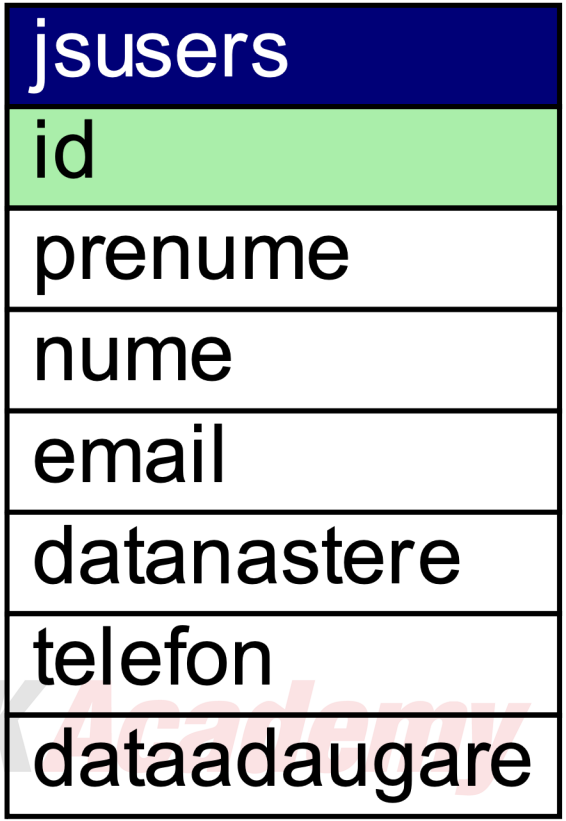


REST API	URL
GET	http://localhost:3002/users
GET	http://localhost:3002/users/id
POST	http://localhost:3002/users
PUT	http://localhost:3002/users/id
DELETE	http://localhost:3002/users/id

ReactCRUD

Backend – MySQL

- Pornim serverul de mysql
- Configuram baza de date: db.sql

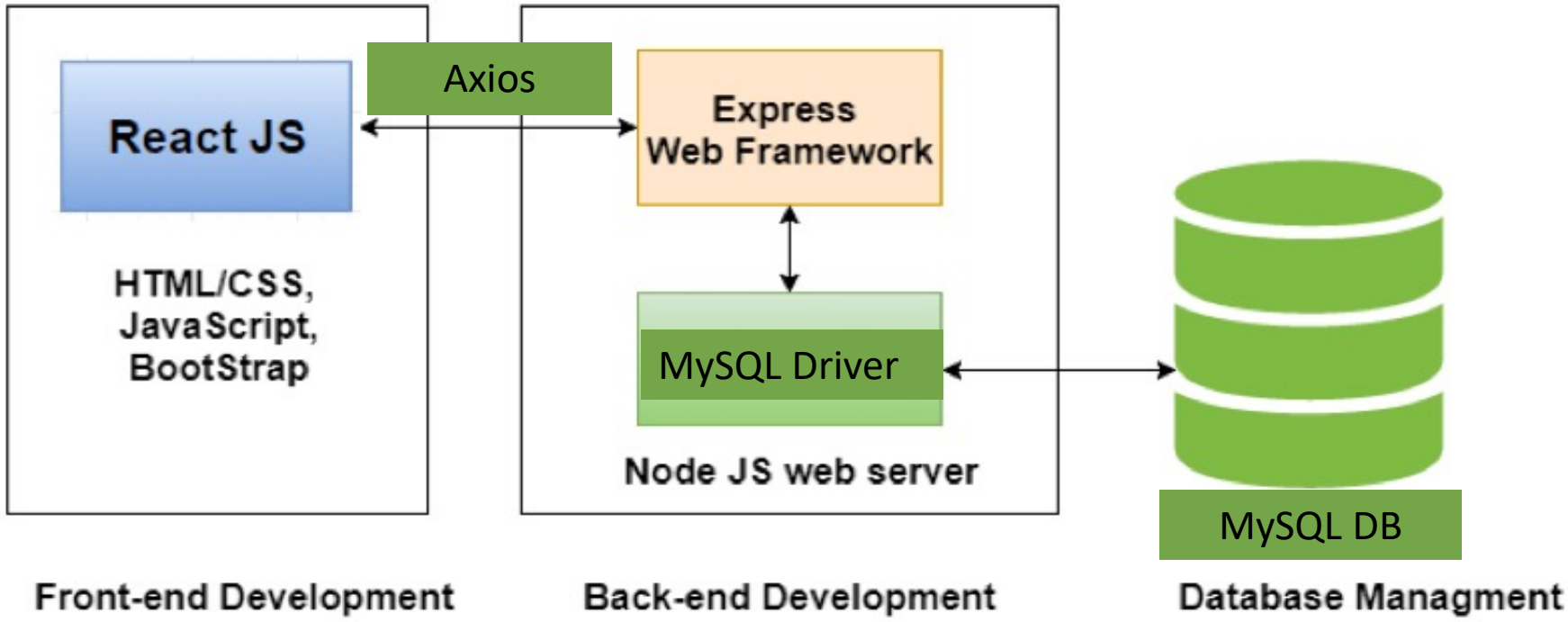


ReactCRUD

Backend

- https://github.com/aadiaconitei/jsdevapp2023/tree/master/angularCRUD/backend
- *npm install*
- *npm run dev*
- http://localhost:3002

ReactCRUD



ReactCRUD

- ✓ npm i create-react-app -g
- ✓ mkdir reactCRUD
- ✓ cd reactCRUD
- ✓ npm view react version
- ✓ **create-react-app** frontend
- ✓ **cd** frontend
- ✓ **npm start**

ReactCRUD

- ✓ /public/index.html
- ✓ /src/index.js
- ✓ /src/App.js – modificăm acest fișier

ReactCRUD

Module necesare

- ✓ **React-Bootstrap** și bootstrap pentru UI.
- ✓ **React-Router-Dom** vă permite să implementați rutarea într-o aplicație React.
- ✓ **Axios** este un client HTTP node.js pe bază de promisiune și se utilizează pentru solicitare de rețea.
- ✓ **Formik** pentru a construi formulare în React.
- ✓ **Yup** pentru validarea formularelor.
- ✓ **Date-fns** pentru formatarea datelor calendaristice

ReactCRUD

Urmăm instrucțiunile din fișierul comenzi.md

Resurse

- ✓ https://www.reactshark.com/blog/guide-react-date-format
- ✓ https://www.geeksforgeeks.org/how-to-build-a-basic-crud-app-with-node-js-and-reactjs/
- ✓ https://create-react-app.dev/docs/deployment/
  https://github.com/bobangajicsm/react-portfolio-website