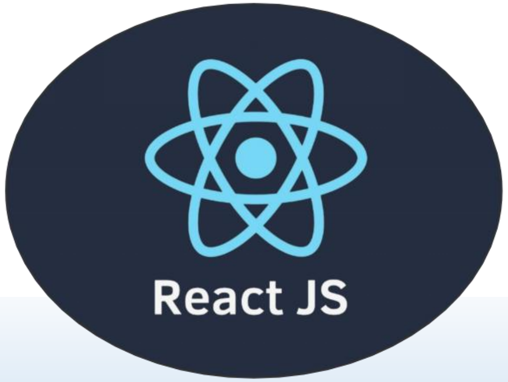


What is React?

Objectives

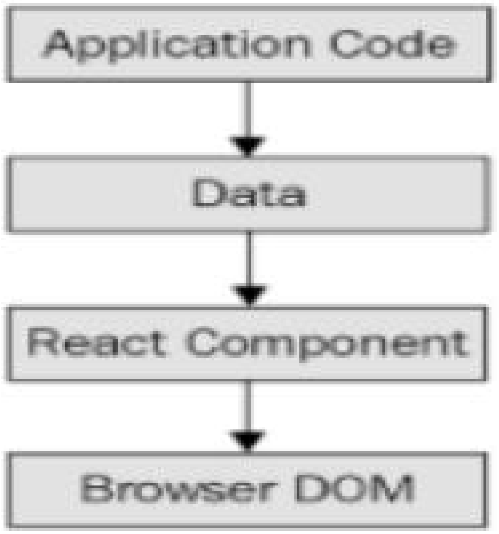
- Overview React
- Describe new in React 18
- Describe installing required dependencies
  - Visual Studio Code
  - Node JS
  - Chrome DevTools
  - React Developer Tools
- Demo Create React App
- Set up a Git

What is React?

- React is "A JavaScript library created by Facebook for building user interfaces (UI)” and a tool for building UI components.
- It allows developers to create reusable UI components and manage the state of an application efficiently.
- Use ways to build dynamic web and mobile applications.
- If React isn't a framework, then what is it exactly?
  - React is just the view layer
  - Simplicity is good
  - Declarative UI structures
  - Data changes over time
  - Performance matters
  - The right level of abstraction

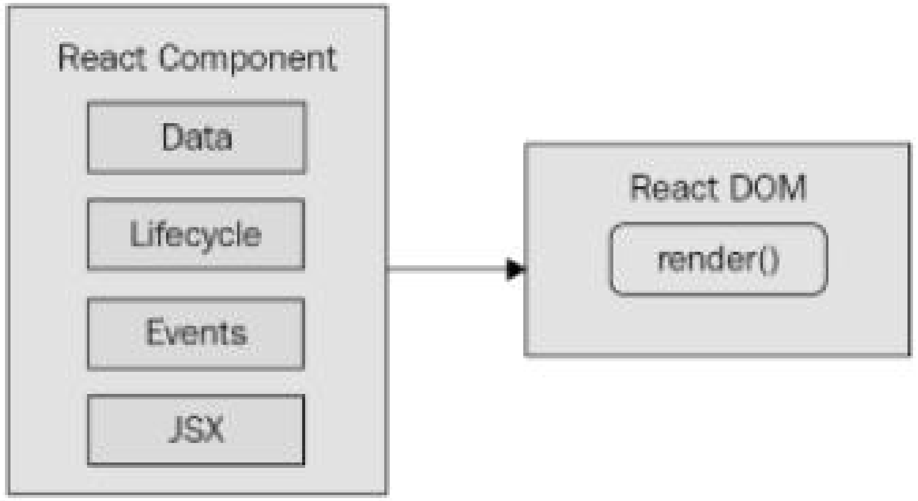
React is just the view layer

- React is generally thought of as the view layer in an application.
- To render this data to the UI, pass it to a React Component, which handles the job of getting the HTML into the page.


Application Code
Data
React Component
Browser DOM

Simplicity is good

- React is divided into two major APIs:
  - **The React Component API**: These are the parts of the page that are rendered by the React DOM.
  - **React DOM**: This is the API that's used to perform the rendering on a web page.
- Within a React component, we have the following areas to think about:
  - Data
  - Lifecycle
  - Events
  - JSX

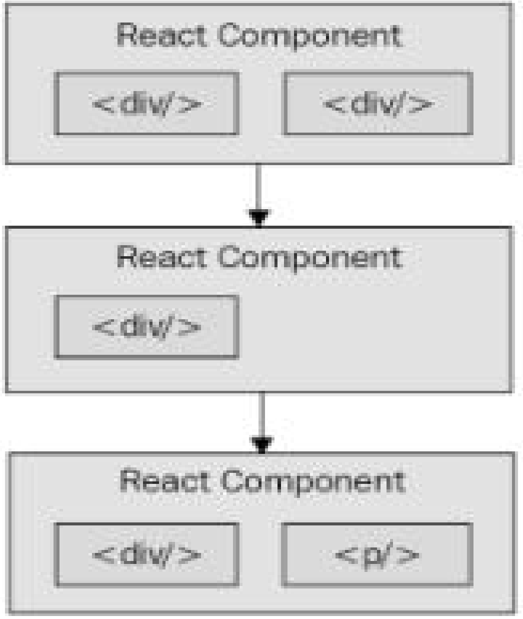


Declarative UI structures

- The syntax used by React components is called JSX (JavaScript XML).
- What's groundbreaking about the declarative JSX approach is that we don't have to perform little micro-operations to change the content of a component.
- The XML-style syntax makes it easy to describe what the UI should look like – that is, what are the HTML elements that this component is going to render? This is called declarative programming and is very well suited for UI development.

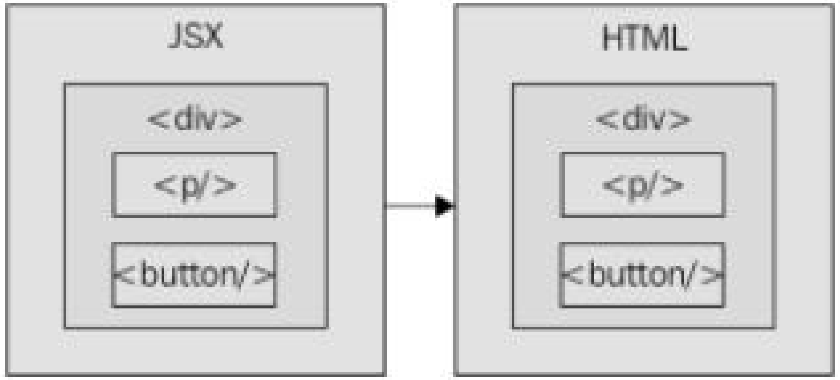
Data changes over time

- JSX is like a static string, representing a chunk of rendered output.
- React components rely on data being passed into them. This data represents the dynamic parts of the UI – for example, a UI element that's rendered based on a Boolean value could change the next time the component is rendered.
- React can handle the performance demands of this approach.

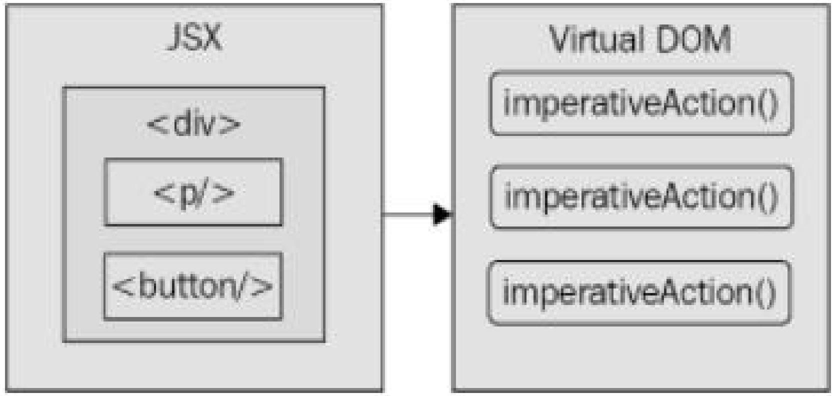


Performance matters

- Can declare the structure of the UI with JSX. This is less error-prone than the imperative approach of assembling the UI piece by piece. However, the declarative approach does present a challenge –performance.
- The Document Object Model (DOM) represents HTML in the browser after it has been rendered. The DOM API is how JavaScript is able to change content on a page.
- Diffing and patching.



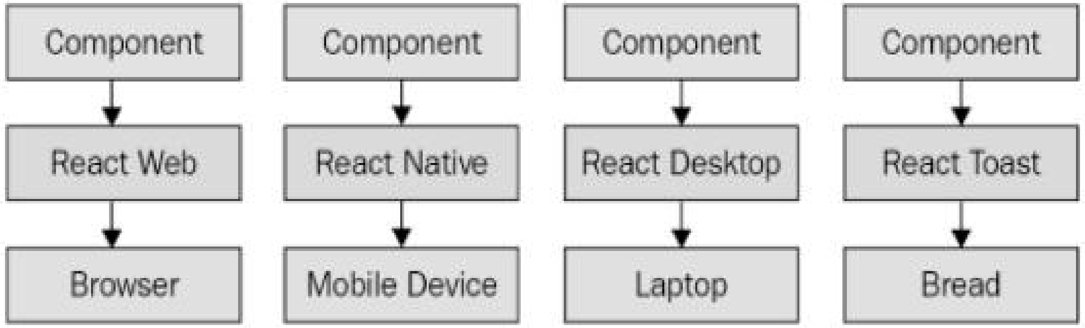
How JSX syntax translates to HTML in the browser DOM



React transpiles JSX syntax into imperative DOM API calls

The right level of abstraction

- React code is abstraction
- From left to right, we have React Web (just plain React), React Native, React Desktop, and React Toast. The same pattern applies:
  - Implement components specific to the target.
  - Implement a React renderer that can perform the platform-specific operations under the hood.



React abstracts the target rendering environment from the components that we implement

Key Features of React

- **Component-Based**: React encourages the creation of modular UI components, making code more organized and maintainable.
- **Virtual DOM**: React uses a virtual representation of the DOM, optimizing updates and improving performance.
- **Large Community**: A vast community of developers and libraries support React, offering solutions to various challenges.

What's new in React 18?

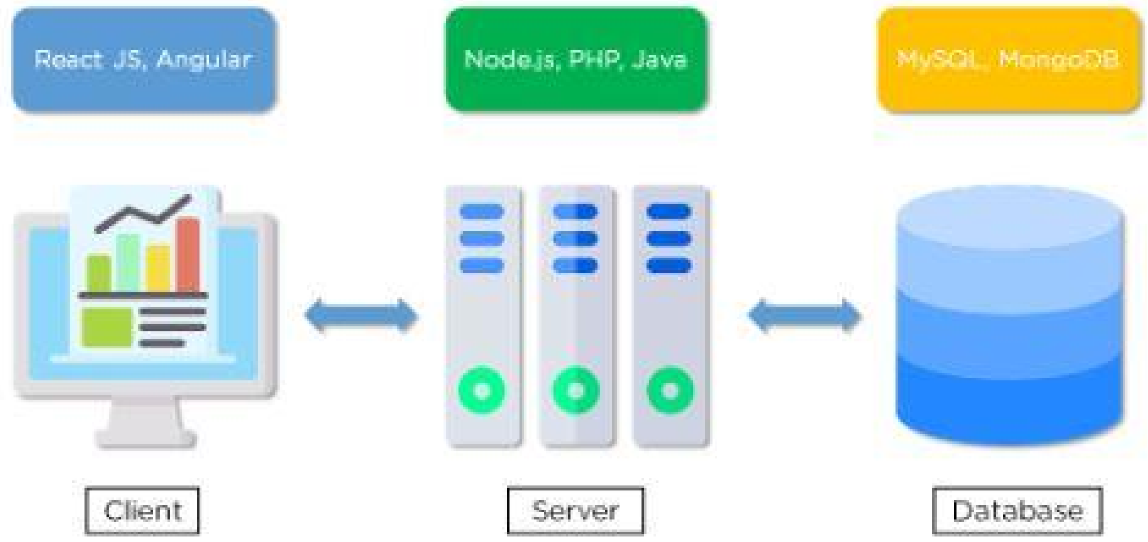
- **Automatic batching**
  - Batching state updates together drastically improves the performance of React applications because it reduces the number of renders to be performed
- **State transitions**
  - The idea with state transitions is that the less important state updates that take place in application should have lower priority than state updates that should happen immediately.

Installing required dependencies

- Visual Studio Code
  - Open https://code.visualstudio.com in your web browser and click the download link for your operating system
- Node.js
  - Visit the npm Package Repository at https://nodejs.org/
- Chrome Dev Tools: Open your Chrome browser
- React Developer Tools
  - Go to the Chrome Web Store at https://chrome.google.com/webstore using your Chrome browser

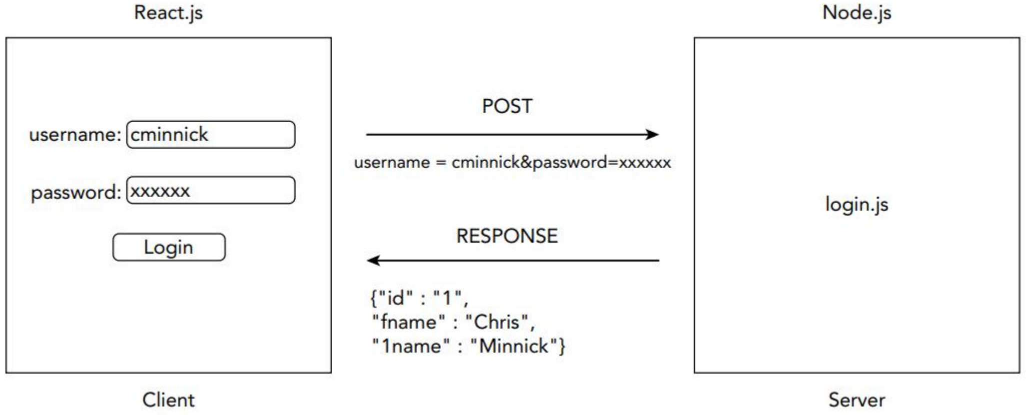
Web Applications

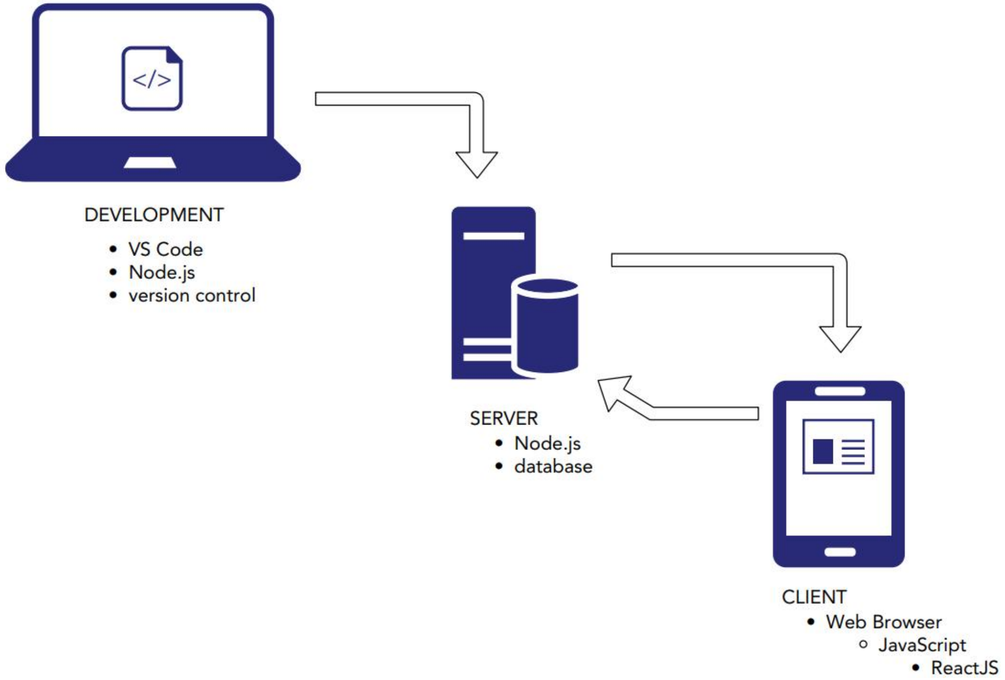
- A web application is a program that runs on a server and is rendered by a client browser, using the internet to access all the resources of that application. It usually can be easily broken down into three parts:
  - Client
  - Server
  - Database

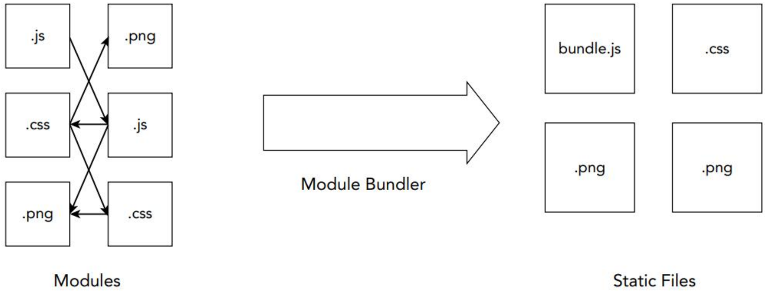


Node.JS

- A way to run JavaScript on web servers
- Common tasks that take place in development and that can be aided by Node.js include
  - Minification
  - Transpiling
  - Module bundling
  - Package management
  - CSS preprocessor
  - Testing frameworks
  - Build automation



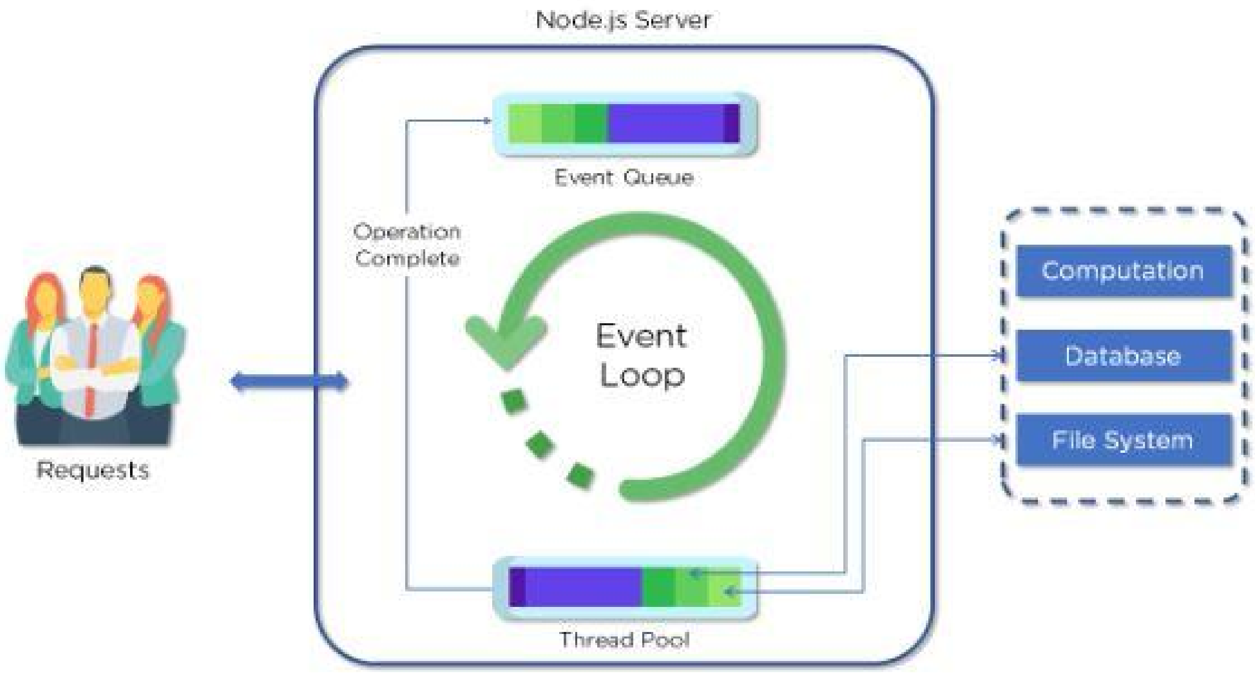




Node Architecture

Parts of the Node.js Architecture:

- Requests
- Node.js Server
- Event Queue
- Thread Pool
- Event Loop
- External Resources



Node Package Manager

- Node package manager (NPM): manages ecosystem of node modules / packages
- A package contains:
  - JS files
  - package.json (manifest)
- npm is the standard package manager for Node.js.

package.json

- A package.json file affords you a lot of great things:
  - It serves as documentation for what packages your project depends on.
  - It allows you to specify the versions of a package that your project can use using semantic versioning rules.
  - Makes your build reproducible, which means that its way easier to share with other developers.
  - Source: https://docs.npmjs.com/getting-started/using-a-package.json

Initializing package.json

- To initialize a package.json file for your project, type at the prompt in your project directory:

  npm init / npm init -y

- If a project has a package.json file, by running

  npm install

- Can install a specific package by running

  npm install <package-name>

- Ex: *npm i bootstrap@5.3.1*

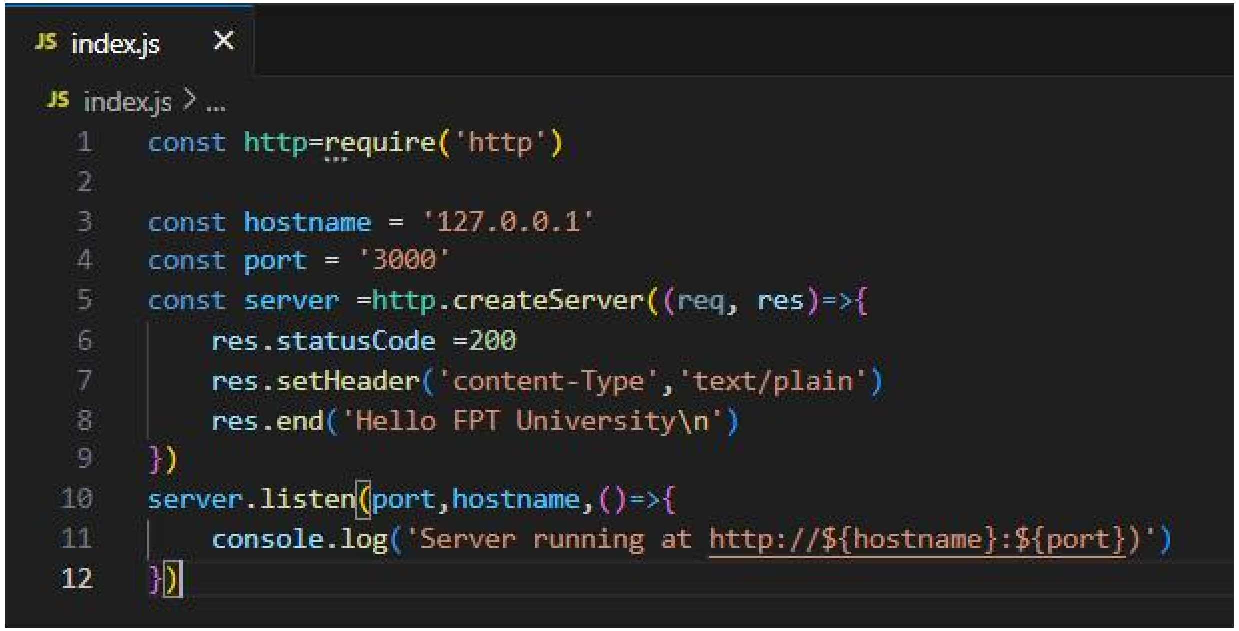
Using npm

- As for the optionalDependencies the difference is that build failure of the dependency will not cause installation to fail.
- See more flags added to this command:
  - **--save -dev** installs and adds the entry to the package.json file devDependencies
  - **--no-save** installs but does not add the entry to the package.json file dependencies
  - **--save-optional** installs and adds the entry to the package.json file optionalDependencies
  - **--no-optional** will prevent optional dependencies from being installed
  - **--save --force** attempting to install or update dependencies that may cause conflicts or compatibility issues

Exercise 1: Install Node.js

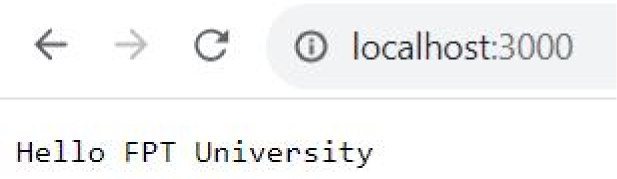
Example Node.js

- The most common example Hello FPT University of Node.js is a web server:

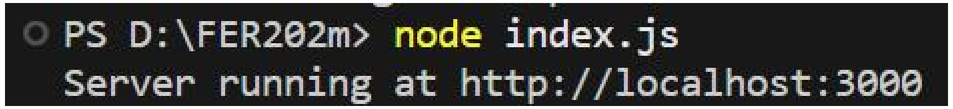


```
const http=require('http')

const hostname = '127.0.0.1'
const port = '3000'
const server =http.createServer((req, res)=>{
    res.statusCode =200
    res.setHeader('content-Type','text/plain')
    res.end('Hello FPT University\n')
})
server.listen(port,hostname,()=>{
    console.log('Server running at http://${hostname}:${port})')
})
```



```
Hello FPT University
```



```
PS D:\FER202m> node index.js
Server running at http://localhost:3000
```

How much JavaScript to use Node.js?

- Lexical Structure
- Expressions
- Data Types
- Classes
- Variables
- Functions
- this operator
- Arrow Functions
- Loops
- Scopes
- Arrays
- Template Literals
- Strict Mode
- ECMAScript 2015 (ES6) and beyond
- Asynchronous programming and callbacks
- Timers
- Promises
- Async and Await
- Closures
- The Event Loop

Exercise 2: Demo Create new React App

Demo Create new React App

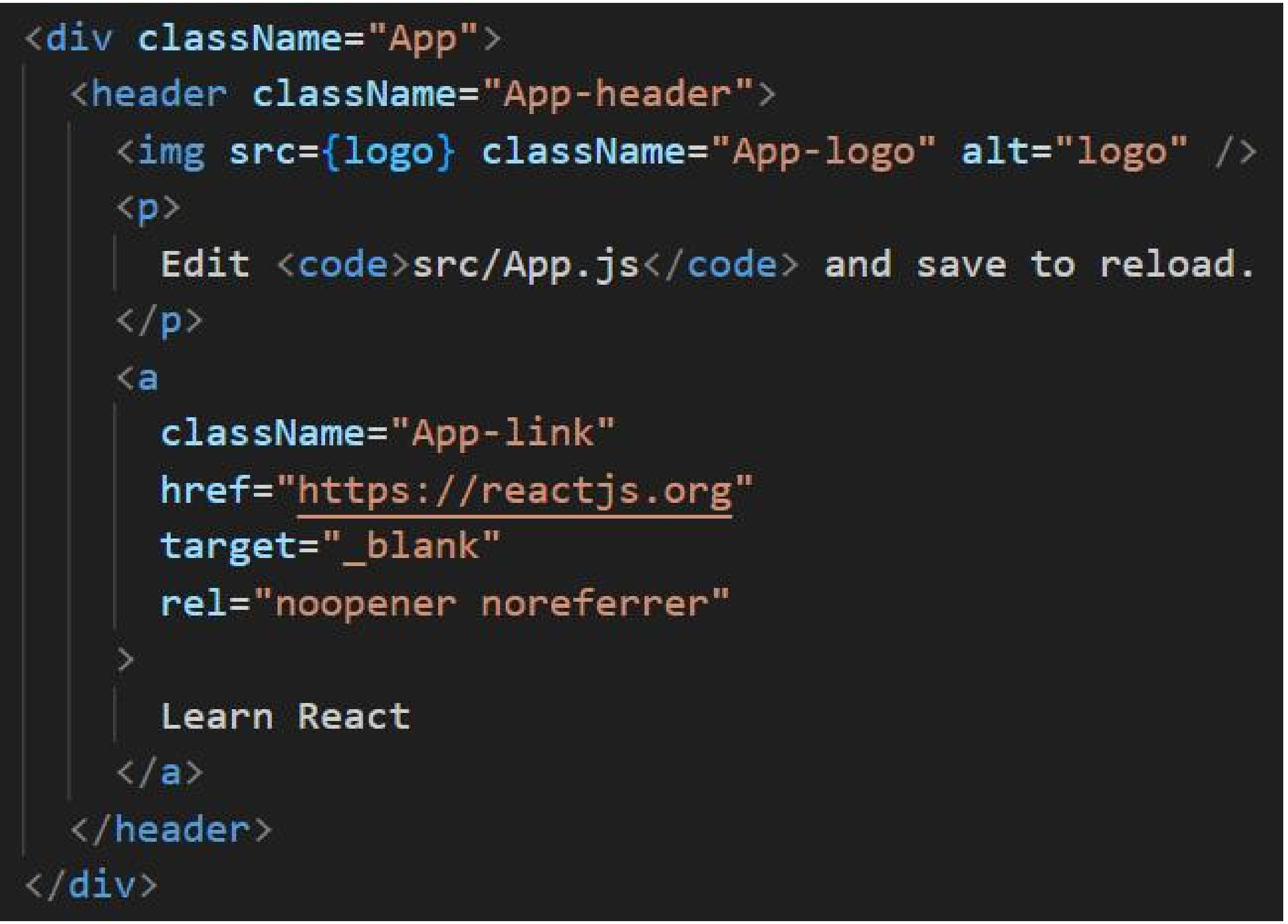
- To create a new React app using Create React App, use the npx command, followed by create-react app, followed by a name that you want to give your new React app.

  npx create-react-app my-new-app

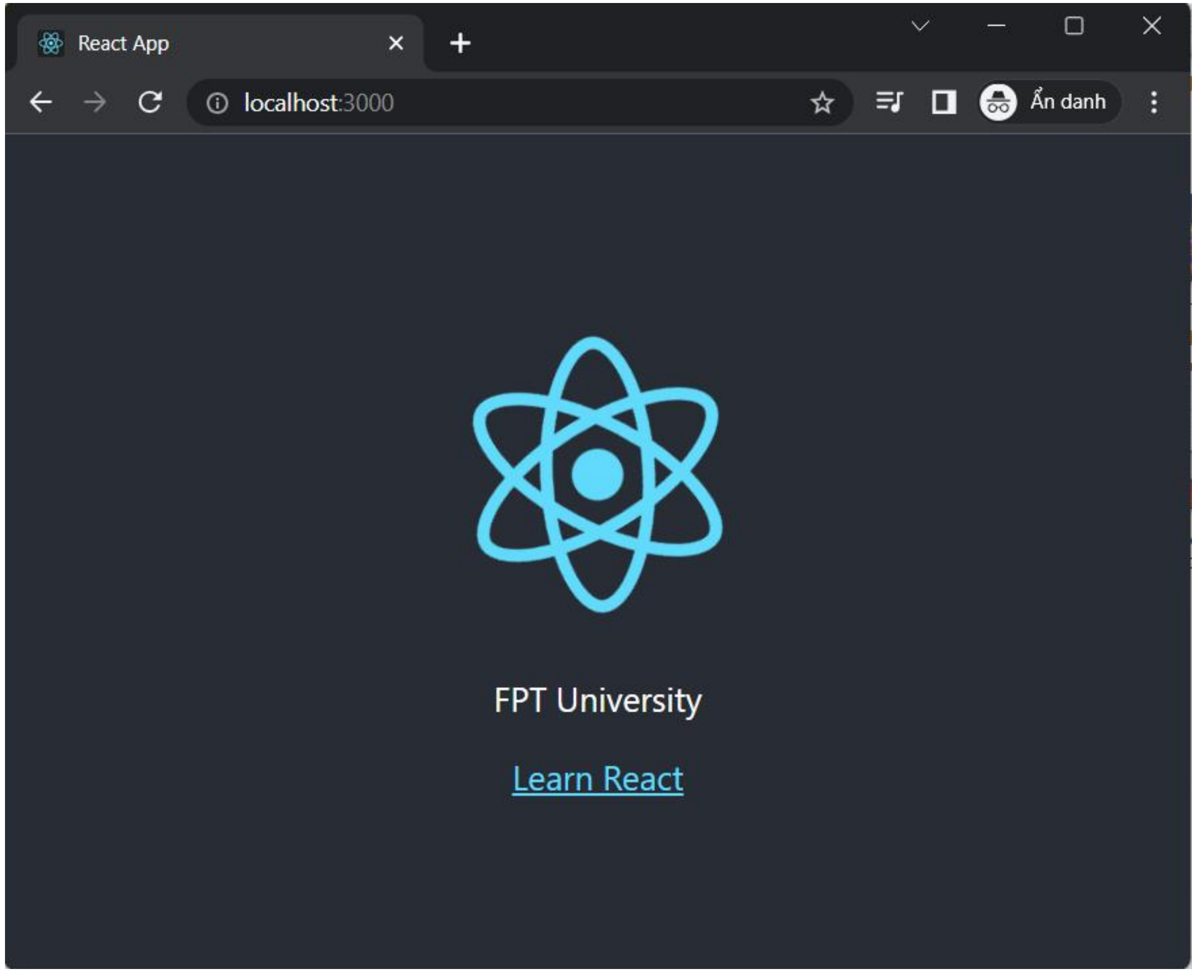
  cd my-new-app

- **Note:**
  - Naming Your React App
  - Making Your First React App

Run React App



```
<div className="App">
  <header className="App-header">
    <img src={logo} className="App-logo" alt="logo" />
    <p>
      Edit <code>src/App.js</code> and save to reload.
    </p>
    <a
      className="App-link"
      href="https://reactjs.org"
      target="_blank"
      rel="noopener noreferrer"
    >
      Learn React
    </a>
  </header>
</div>
```



FPT University

Learn React

Git

Some Basic Concepts

- Version Control: software tool(s) that enable the management of changes to source code
  - Maintaining version history
- Several version control tools: CVS, SVN, Git etc.
- Distributed version control system
- Developed by Linus Torvalds for managing Linux kernel development
- Widely adopted now by several projects
  - The Node ecosystem thrives on it

Git Install

- Download Git to your computer at: https://git-scm.com/downloads
- Choose the Git version that is suitable for the operating system you are using on your computer and download it.
- Open the installation file and run the setup program.
- Agree to the terms and continue the installation process.
- Choose the installation options that you want to use and continue the installation process.
- Complete the installation process and restart your computer if required.
- Open Command Prompt or Terminal on your computer and check if Git has been successfully installed by entering the following command:

  ```
  git --version
  ```

- If Git has been installed successfully, the version of Git will be displayed on Command Prompt or Terminal.

Online Git Repository

- Several online Git repository service providers:
  - GitHub (https://github.com)
  - Bitbucket (https://bitbucket.org)

Config Git

- Configure Git

  git config --global user.name "yourname "

  git config --global user.email "youremail"

- Creating Git Folder

  cd myproject

- Initialize Git

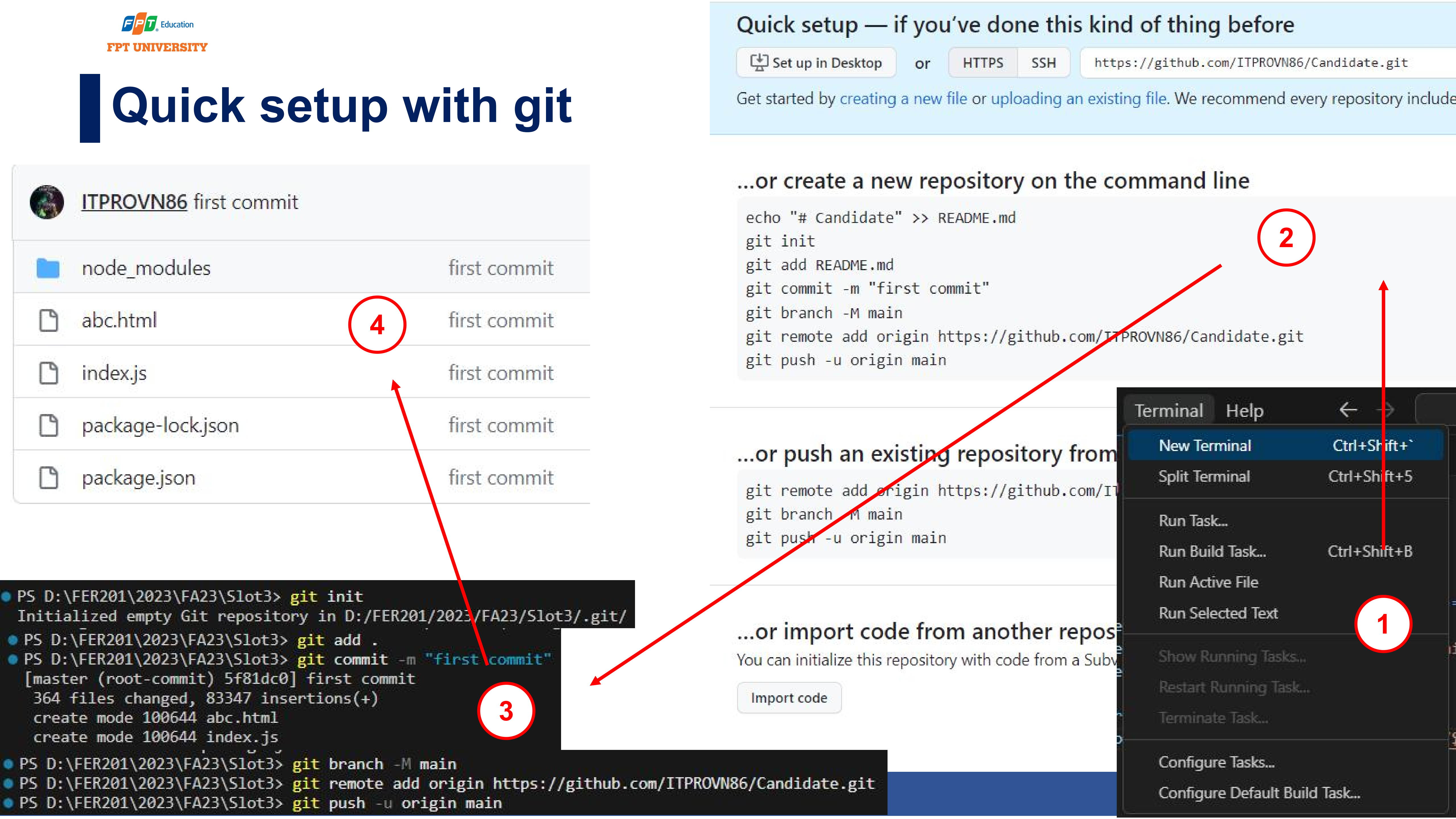
  git init

Using Git

- Add the remote online repository
  git remote add origin <repository URL>
- Git Staging Environment
  git add ‘your file name’
  git add --all
- Check status
  git status
- Git Commit
  git commit -m "First of Hello World!"
- Git Commit Log
  git log
- New Git Branch
  git branch ‘hello-world’
- Git Checkout
  git checkout hello-world
- Push Changes to GitHub
  git push -u origin master

FPT Education
FPT UNIVERSITY

Quick setup with git



ITPROVN86 first commit

node_modules	first commit
abc.html	first commit
index.js	first commit
package-lock.json	first commit
package.json	first commit

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/ITPROVN86/Candidate.git

Get started by creating a new file or uploading an existing file. We recommend every repository include

...or create a new repository on the command line

```
echo "# Candidate" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ITPROVN86/Candidate.git
git push -u origin main
```

...or push an existing repository from

```
git remote add origin https://github.com/IT
git branch -M main
git push -u origin main
```

...or import code from another repos

You can initialize this repository with code from a Subv

Import code

```
PS D:\FER201\2023\FA23\Slot3> git init
Initialized empty Git repository in D:/FER201/2023/FA23/Slot3/.git/
PS D:\FER201\2023\FA23\Slot3> git add .
PS D:\FER201\2023\FA23\Slot3> git commit -m "first commit"
 [master (root-commit) 5f81dc0] first commit
  364 files changed, 83347 insertions(+)
  create mode 100644 abc.html
  create mode 100644 index.js
PS D:\FER201\2023\FA23\Slot3> git branch -M main
PS D:\FER201\2023\FA23\Slot3> git remote add origin https://github.com/ITPROVN86/Candidate.git
PS D:\FER201\2023\FA23\Slot3> git push -u origin main
```

Exercise 3: Install, Setting up and Push Code to Git

Summary

- Concepts were introduced:
    - Overview React
    - Describe new in React 18
    - Describe installing required dependencies
        - Visual Studio Code
        - Node JS
        - Chrome DevTools
        - React Developer Tools
    - Demo Create React App
    - Set up a Git