

The background of the slide is a light beige color. In the top left corner, there is a corkboard with a few papers pinned to it. In the center, there is a large, stylized illustration of a laptop. The laptop screen displays a website layout with a blue header, a main content area with a colorful bar chart, and three grey rectangular boxes at the bottom. To the left of the laptop, there is a red fire alarm pull station with two gauges. Above the laptop, there are several colorful circles connected by dashed lines, representing a network or data flow. The circles are labeled with various web technologies: 'www' (blue), 'HTML5' (red), 'js' (red), 'Cloud' (grey), 'XML' (orange), and 'PHP' (green).

ECAP472

WEB TECHNOLOGIES

Dr. Pritpal Singh

Associate Professor

Learning Outcomes

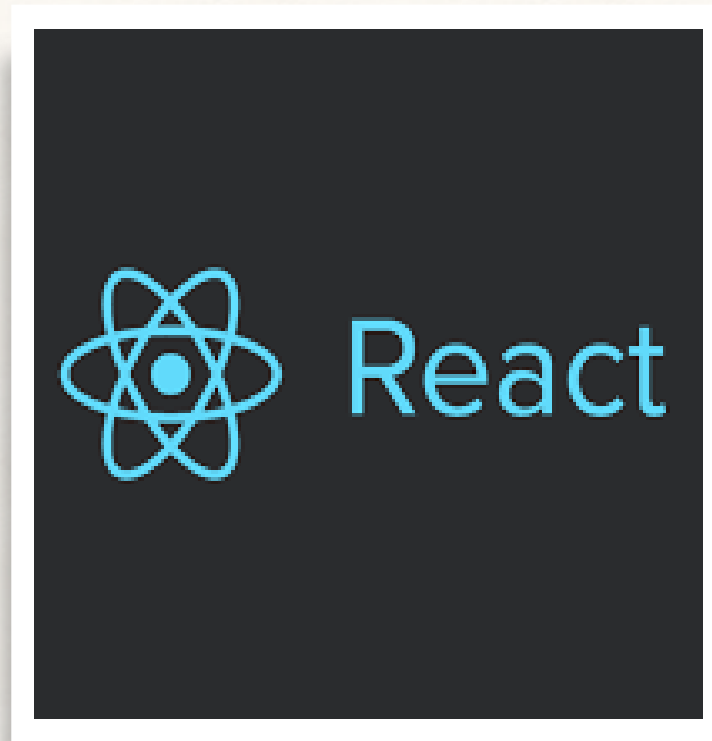


After this lecture, you will be able to

- Understand basics of React folder structure.
- Create first react application.

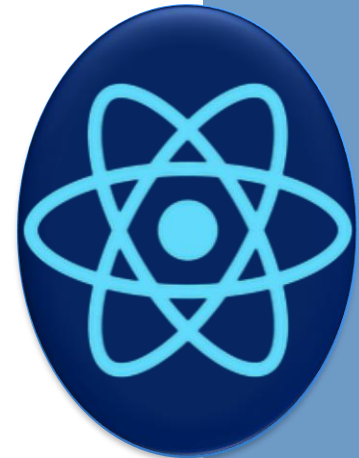
React

React (also known as React.js or ReactJS) is a **free and open-source front-end JavaScript library** for building **user interfaces** based on UI components. It is maintained by **Meta** (formerly Facebook) and a community of individual developers and companies.



React (JavaScript library)

React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.



React installation



Install NodeJS and NPM.

Install Visual Studio Code.

Install React from terminal .

```
npm install -g create-react-app
```

Npm start* npm start



```
Windows PowerShell
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>cd reacttut

C:\Users\Billa\reacttut>cd awesomeapp
C:\Users\Billa\reacttut\awesomeapp>npm start

> awesomeapp@0.1.0 start
> react-scripts start
✓ Something is already running on port 3000.

Would you like to run the app on another port instead? ... yes
(node:18616) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:18616) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

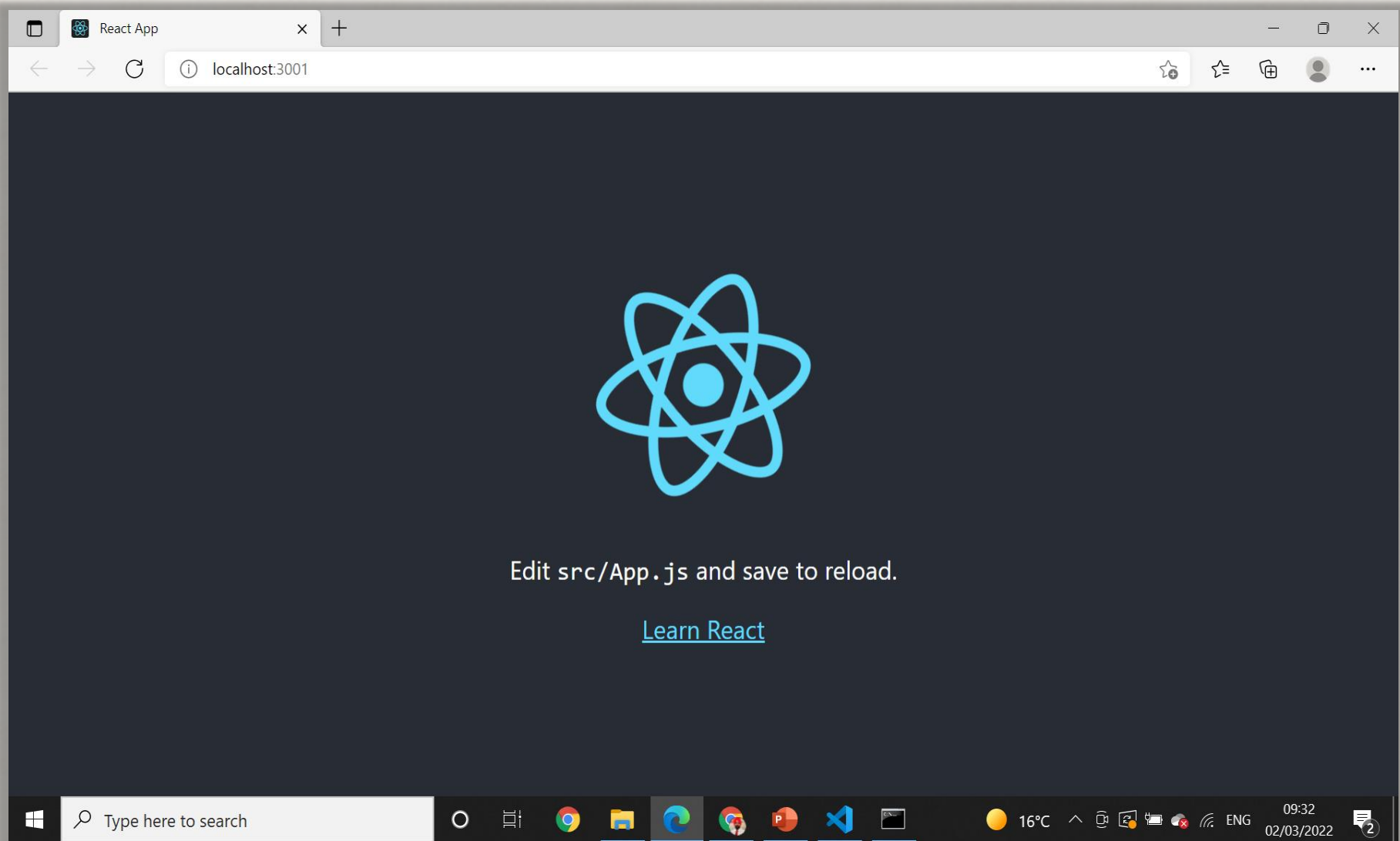
You can now view awesomeapp in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.56.1:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

assets by path static/ 1.49 MiB
  asset static/js/bundle.js 1.48 MiB [emitted] (name: main) 1 related asset
  asset static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js 6.93 KiB [emitted] 1 related asset
  asset static/media/logo.6ce24c58023cc2f8fd88fe9d219db6c6.svg 2.57 KiB [emitted] (auxiliary name: main)
asset index.html 1.67 KiB [emitted]
asset asset-manifest.json 546 bytes [emitted]
cached modules 1.36 MiB [cached] 106 modules
runtime modules 31.3 KiB 15 modules
./node_modules/webpack-dev-server/client/index.js?protocol=ws%3A&hostname=0.0.0.0&port=3001&pathname=%2Fws&logging=none&reconnect=10 6.59 KiB [built] [code generated]
webpack 5.69.1 compiled successfully in 1980 ms
```

Local host .



How does React Work?



- React creates a VIRTUAL DOM in memory.
- Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.
- **React only changes what needs to be changed!**
- React finds out what changes have been made, and changes only what needs to be changed.

React.JS History



- Current version of React.JS is V17.0.2 (August 2021).
- Initial Release to the Public (V0.3.0) was in July 2013.
- **React.JS was first used in 2011 for Facebook's Newsfeed feature.**
- Facebook Software Engineer, Jordan Walke, created it.
- Current version of create-react-app is v4.0.3 (August 2021).
- create-react-app includes built tools such as webpack, Babel, and ESLint.

How to use files in public folder in ReactJS ?

- The public folder contains static files such as index.html, JavaScript library files, images, and other assets, etc. which you don't want to be processed by webpack. Files in this folder are copied and pasted as they are directly into the build folder. Only files inside the `public` folder can be referenced from the HTML.

React folder structure



```
✓ MY-REACT-APP
> node_modules
> public
✓ src
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
  reportWebVitals.js
  setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

After creating a React application

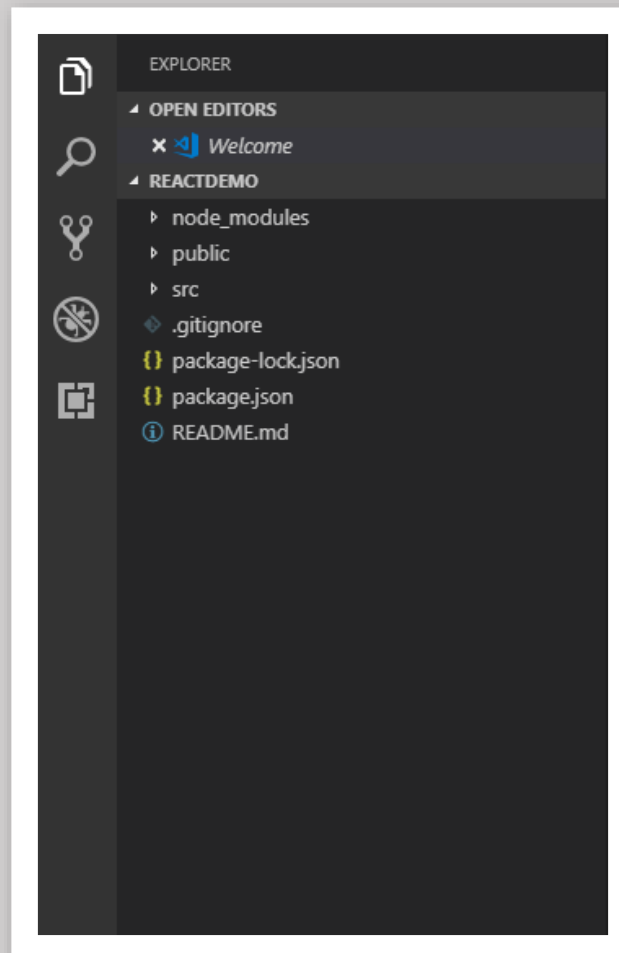


- After creating a React application, we will notice that in Terminal, it provides quick tips or commands that will help in executing the React application. For example :-
- **npm start** - This command will start the development server.
- **npm run build** - Bundles the app into static files for production.
- **npm test** - starts the test runner.
- **npm run object** - Removes this tool and copies build dependencies, configuration files, and scripts into the app directory. If you do this, you can't go back!

Folder Structure



The React application automatically creates required folders, as shown below.



.gitignore



- .gitignore - This file is the standard file which is used by source control tool git to identify which files and folders are need to be ignored while committing the code. **Until and unless this file exists, the create-react-app command will not create a git repo in this folder.**

```
1 # See https://help.github.com/articles/ignoring-files/ for more about ignoring files.
2
3 # dependencies
4 /node_modules
5 /.pnp
6 .pnp.js
7
8 # testing
9 /coverage
10
11 # production
12 /build
13
14 # misc
15 .DS_Store
16 .env.local
17 .env.development.local
18 .env.test.local
19 .env.production.local
20
21 npm-debug.log*
22 yarn-debug.log*
23 yarn-error.log*
24
```

package.json



- package.json – This file contains dependencies and scripts required for the project.

```
package.json x
1  {
2    "name": "reactdemo",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "react": "^16.8.6",
7      "react-dom": "^16.8.6",
8      "react-scripts": "3.0.1"
9    },
10   "scripts": {
11     "start": "react-scripts start",
12     "build": "react-scripts build",
13     "test": "react-scripts test",
14     "eject": "react-scripts eject"
15   },
16   "eslintConfig": {
17     "extends": "react-app"
18   },
19   "browserslist": {
20     "production": [
21       ">0.2%",
22       "not dead",
23       "not op_mini all"
24     ],
25     "development": [
26       "last 1 chrome version",
27       "last 1 firefox version",
28       "last 1 safari version"
29     ]
30   }
31 }
32
```

Package.json : contains all settings for React applications

- name - points to name of your app.
- version - refers to the current version that the application is using.
- “private”: true - is a foolproof setting which avoids accidentally publishing of your react app as a public package in npm ecosystem.
- Dependencies will contain all required node modules and versions required for the application.



- **package-lock.json** contain exact dependency tree to be installed in **/node_modules**. It helps while a team is working on private apps to ensure that they are working on the same version of dependencies and sub-dependencies. It also maintains a history of changes done in package.json so, that at any point of time, when required previous changes can be looked back in the package-lock.json file.



- This folder contains all dependencies and sub-dependencies specified in package.json used by React app. It contains more than 800 subfolders, this folder is automatically added in the .gitignore file



- This folder contains files which don't require additional processing by webpack. **The index.html file is considered as an entry point for the web application.** Here, the favicon is a header icon and manifest.xml file contains configuration when your application is used for Android app



- This folder is the heart of React application as it contains JavaScript which needs to be processed by webpack. In this folder, there is a main component **App.js**, its related styles (**App.css**), test suite (**App.test.js**), **index.js**, and its style (**index.css**); which **provide an entry point into the App**. Lastly, it contains **registerServiceWorker.js** which takes care of caching and updating files for the end user. It helps in offline capability and faster page loading after the first visit.

Components



- After all this, we add /Component folder in src to add our custom component and its related files and /Views folder for adding React views and its related files.

Practical

That's all for now...