Laboratory Work
Structuring Real-World React Projects

Tasks:
1. Setting Up Your React Development Environment
2. Choosing a Code Editor for React Development
3. Organizing the Project Folder Structure
4. Working with the Terminal in React Development
5. Starting and Running a React Project with Vite
6. React Project Setup Wrap-Up

Step-by-step guide on how to start a real-world React project. From setting up your development environment to understanding project organization and critical tools, you'll gain the knowledge and confidence to embark on your React journey.

1. Setting Up Your React Development Environment

In the previous sections, we have learned the basics of React. We now know how to create components, manage them, render information, and implement logic. However, all these tasks were done in CodeSandbox, which is excellent for studying.

But what about real-world projects? Do all developers create projects in CodeSandbox? Of course not. This section is dedicated to helping you start your project on your local machine, guide through the process step by step.

Step 1: Install Node.js and npm

We need to install Node.js and npm on the local machine.

Node.js is an open-source runtime environment that allows us to run JavaScript code on the server side.

npm stands for "Node Package Manager". It is a package manager for Node.js that allows developers to easily install, manage, and share reusable JavaScript code packages or modules.

- Go to the official Node.js website: https://nodejs.org;
- Download the installer for your operating system (Windows, macOS, or Linux). We are interested in the LTS version as it is stable and working;
- Run the installer and follow the installation instructions;
- After the installation, we should have Node.js and npm (Node Package Manager) installed on the machine.

How can we check do we have Node.js and npm installed?

To check if Node.js and npm are installed on the machine, we can open the computer's terminal (command prompt, PowerShell, or terminal emulator) and run the commands. Steps:

1. Open the terminal;

- **Windows**: Press Win + X and select "Windows PowerShell" from the menu. Alternatively, search for "PowerShell" in the Start menu;

Note. If none of these methods work, it is recommended to refer to the documentation or support resources for the operating system to find the appropriate way to open the terminal. Or we can always google.

2. Check Node.js version;

In the terminal run the following command.

```
node -v
```

This command will display the version of Node.js installed on the machine. If Node.js is installed, we see the version number.

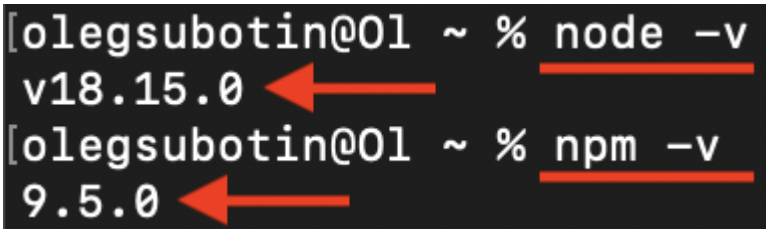
3. Check npm version.

In the terminal run the following command.

```
npm -v
```

This command will display the version of npm installed on the machine. If npm is installed, we see the version number.

If both commands return the version numbers, Node.js and npm are successfully installed and available on the machine. If the commands are not recognized or return an error, it means that Node.js and/or npm are not installed, or their installation paths are not correctly configured. In that case, we may need to revisit the installation steps for Node.js and ensure the installation was completed successfully. Sometimes it is recommended to restart the machine.



Note. What version we get doesn't matter for now.

2. Choosing a Code Editor for React Development

Step 2: Install code editor

To proceed, we need to install a code editor. If you already have a code editor, skip this step and move on to the next chapter. We are free to choose any code editor we find comfortable working with. Some popular options include:

- Visual Studio Code - https://code.visualstudio.com/;
- Sublime Text - https://www.sublimetext.com/;
- WebStorm - https://www.jetbrains.com/webstorm/.

Throughout this course, all examples will be demonstrated using Visual Studio Code. It's important to note that the choice of the editor doesn't significantly impact the content, but the external interface may differ.

Steps to follow:

1. Choose a code editor;
2. Go to the website of the chosen code editor and download the installer for the operating system;
3. Run the installer and follow the installation instructions.

3. Organizing the Project Folder Structure

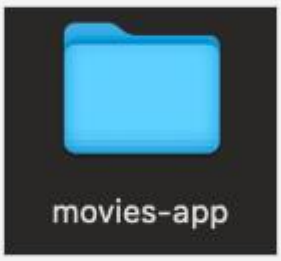
Step 3: Create the project folder

Now that we have all the necessary external tools, we can start creating the project. The task is to create a new empty folder specifically for the project, where nothing else will be stored.

Steps to follow:

1. **Choose a Location**: Select a location on your machine where you want to create the project folder. It can be anywhere you prefer, such as your desktop or a specific directory;
2. **Create a New Folder**: Right-click in the chosen location and select "New Folder" from the context menu;
3. **Name the Folder**: Give the folder a meaningful name. This will serve as the root folder for your React project, so choose an informative and relevant name for your project's purpose. For example, you can name it `movies-app`,`store-app`, `posts-app`, `learning-system-app`, etc.

For example let's create the empty folder with the name `movies-app`.

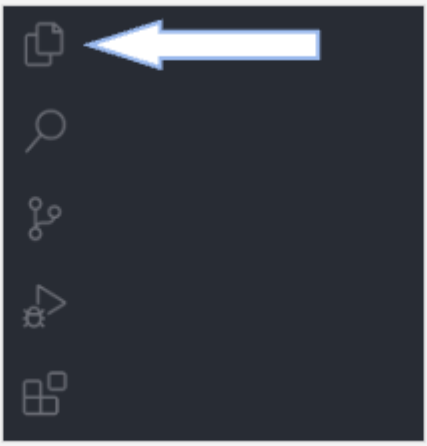


Step 4: Open the Folder in Your Code Editor

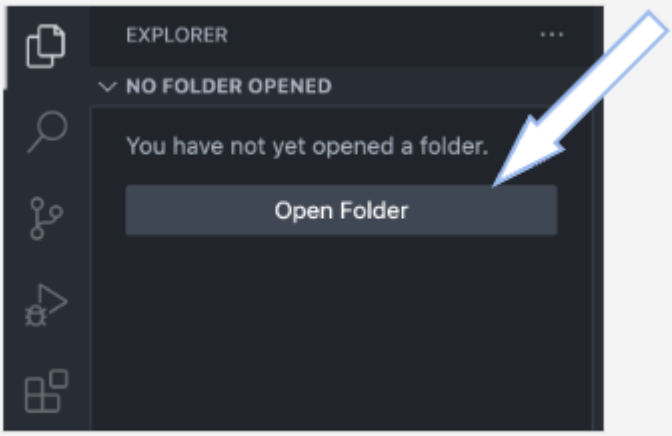
To start working on the project, you need to open the `movies-app` folder in your code editor, whether it's Visual Studio Code or another editor of your choice.

Follow these steps:

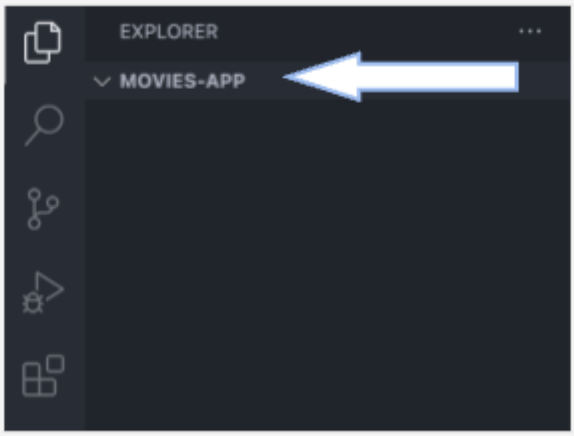
- Launch your code editor. In our example, we'll use Visual Studio Code (VS Code);
- Click on the explorer icon located in the top-left corner of the VS Code interface;



- Click the "Open Folder" button in the explorer panel;



- Look for the movies-app folder on the machine and select it. After completing these steps, you should see the folder name, movies-app, displayed in your code editor's interface.



Now that your project folder is set up and opened in your code editor, you're ready to start building your React application!

4. Working with the Terminal in React Development

Step 5: Open terminal in the code editor

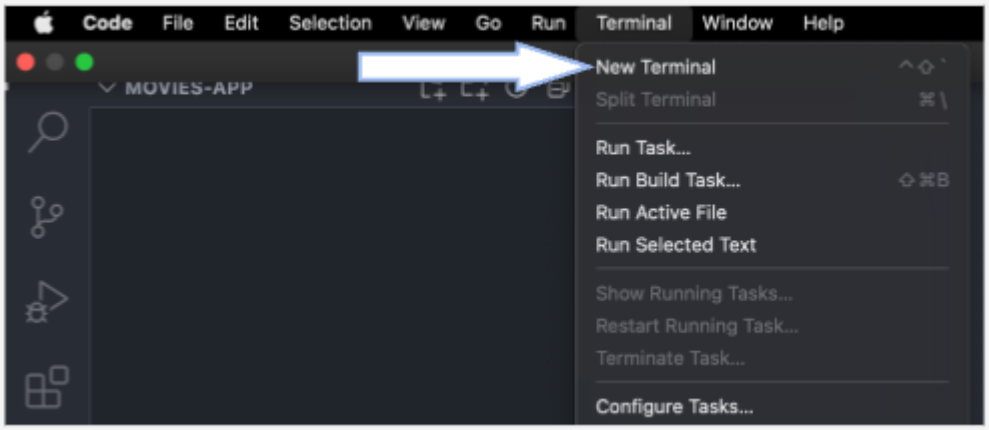
Now that we have all the necessary tools and have opened the project folder in the code editor let's open the terminal within the code editor. In VS Code, this process is straightforward.

Steps to Follow:

- Find the word Terminal in the top bar of VS Code;



- Click on it to open a dropdown menu;

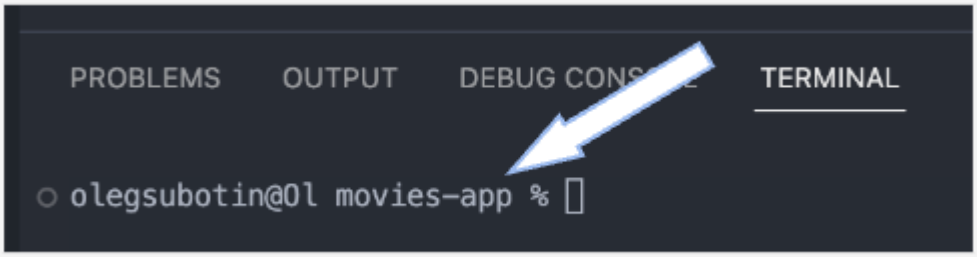


- Select New Terminal from the dropdown menu.

After completing these steps, the terminal will open at the bottom of the VS Code window. Please note that the appearance may vary slightly depending on your operating system. Make sure that the project folder's name is visible in the terminal.

- Go to the folder (type the command in the terminal window):

```
cd  C:\movies-app
```

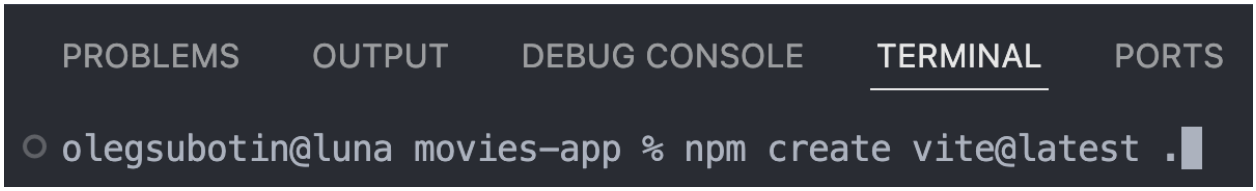


5. Starting and Running a React Project with Vite

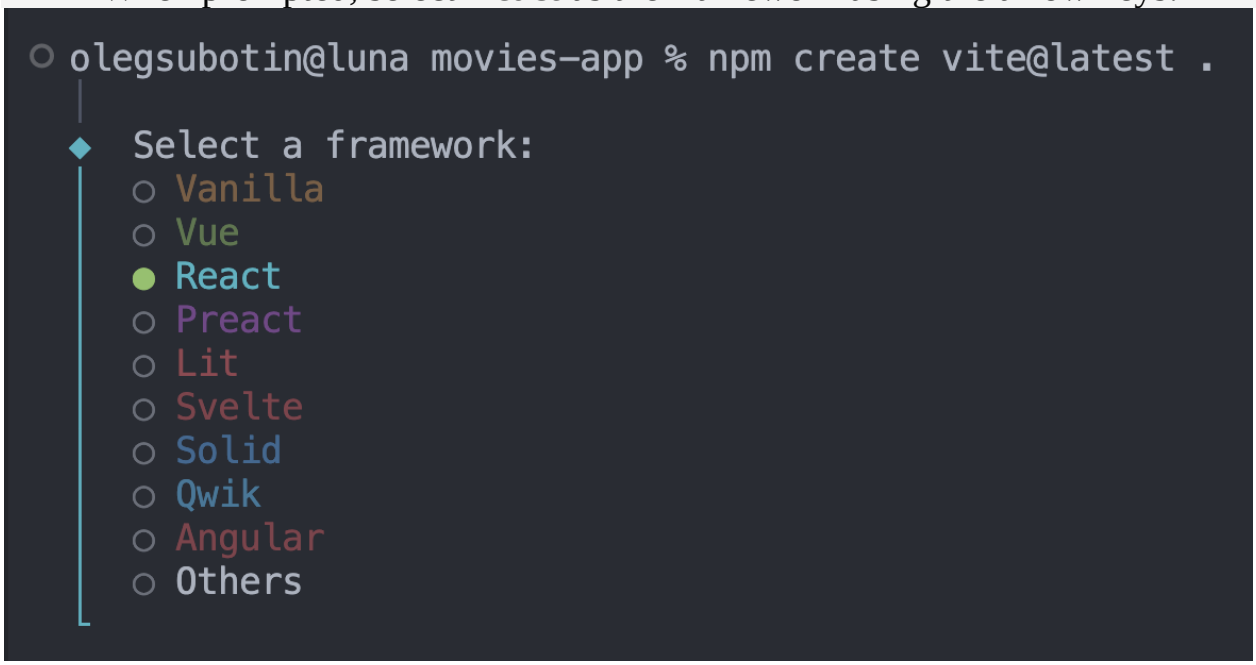
Step 6: Initialize the Project

To set up the project, use the terminal. Open the terminal in your code editor (if it's not already open), run the following command, and press Enter:

```
npm create vite@latest
```



When prompted, select **React** as the framework using the arrow keys:



Then, choose either **JavaScript** or **TypeScript**, depending on your preference:

```
olegsubotin@luna movies-app % npm create vite@latest .

◇  Select a framework:
   React

◆  Select a variant:
   ○ TypeScript
   ○ TypeScript + SWC
   ● JavaScript
   ○ JavaScript + SWC
   ○ React Router v7 ↗
   ○ TanStack Router ↗
```

After the setup is complete, Vite will prompt you to install dependencies. Run the following command:

```
npm install
```

```
olegsubotin@luna movies-app % npm install

added 153 packages, and audited 154 packages in 10s

32 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Once the installation is done, start the development server by running:

```
npm run dev
```

```
olegsubotin@luna movies-app % npm run dev

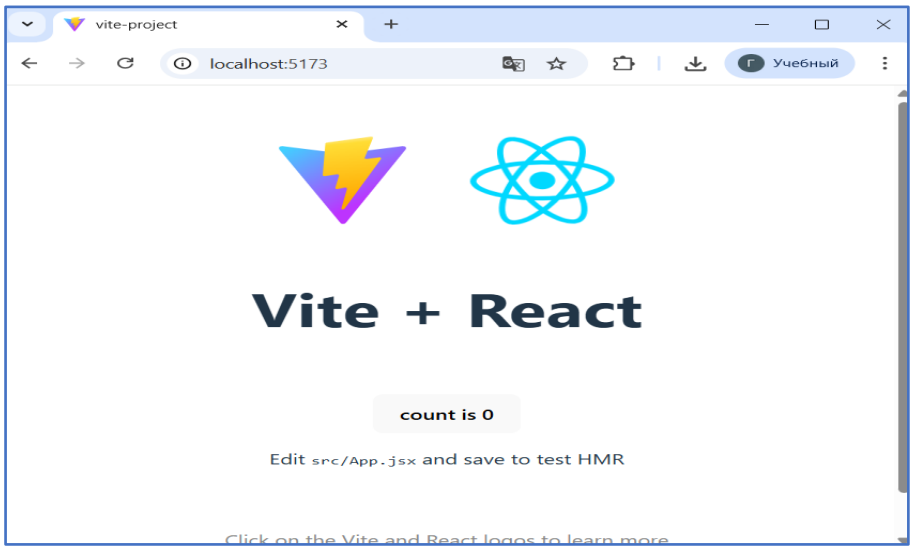
> movies-app@0.0.0 dev
> vite


  VITE v6.3.3  ready in 384 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

After running these commands, you'll see a local development URL — typically:

http://localhost:5173/. You can open this link in your browser to view the app live.



Step 7: Begin Customizing

The project is set up, you can begin customizing it:

- Your main work will happen inside the **src** folder;
  - You'll find default files like App.jsx and main.jsx, which you can modify or replace based on your project's needs;
  - If you worked previously in CodeSandbox, recreate your familiar structure: ensure you have a main.jsx (the Vite equivalent of index.js) as the entry point and build your main application in App.jsx;

- Create additional folders as needed to organize your components, styles, and assets.
- Copy the source files of your React application (for example, from CodeSandbox) into a newly created project. Then, launch the project in your local development environment and verify that it runs correctly in the browser.

6. React Project Setup Wrap-Up

Follow these steps to develop a React app on your local machine:

Install Node.js and npm:

- Visit the official Node.js website at https://nodejs.org;
- Download the installer for your operating system and follow the installation instructions to install Node.js and npm.

Install a Code Editor:

- Choose a code editor that suits your preference, such as Visual Studio Code;
- Install the code editor on your computer;
- This code editor will be your tool for writing React code.

Create a Project Folder:

- Create an empty folder on your computer to store your React project;
- This folder will serve as the root directory for your project.

Open the Code Editor:

- Launch the code editor of your choice;
- Navigate to the project folder you created in step 3;
- This will allow you to start working on your React project within the code editor.

Open the Terminal:

- Within the code editor, open the terminal or command line interface;
- This is where you'll run commands to set up and manage your React project.

Set Up the Project:

- In the terminal, run the command `npx create-react-app .`;
- This command initializes a new React project using the Create React App tool.

Start the Live Page:

- Run the command `npm start` once the project is created;
- This starts the development server and opens a live preview of your React app in the browser.

Customize the Project:

- Open the `src` folder within your project directory;
- You can delete all default files created by Create React App;
- Build your React components, add styles, and implement functionality within the `src` folder.