

# Workshop Technology

# JavaScript Frameworks

# What is a framework?

- A framework is a library that offers **opinions** about how software gets built.
- These opinions allow for **predictability** and **homogeneity** in an application.
  - Predictability allows software to scale to an enormous size and still be maintainable.
  - Predictability and maintainability are essential for the health and longevity of software.

# Vanilla JS or Framework?

- Vanilla JS
  - A less complex site, for example a personal to-do list or a site that displays mostly static content
- Framework
  - A large site with a complex UI. Frameworks provide solutions to common problems that would take an absurd amount of time and patience to implement with pure JavaScript.

# Stack-Overflow survey 2025

- Stack-Overflow is a popular Q&A website for programmers.
- Every year, they conduct a survey to gather insights about the developer community.
- Here are the [results](#).

# React

# Get started with React (1/2)

- Open CMD/Terminal
- `pnpm create vite@latest`
  - Project name : vite-project
  - Select a framework : React
  - Select a variant : JavaScript + SWC
  - Use rolldown-vite...? : No
  - Install with pnpm...? : Yes

```
admin@work ~\Desktop pnpm create vite@latest
.../19add56e802-4694 | +1 +
.../19add56e802-4694 | Progress: resolved 1, reused
|
◇ Project name:
vite-project
|
◇ Select a framework:
React
|
◇ Select a variant:
JavaScript + SWC
|
◇ Use rolldown-vite (Experimental)?:
No
|
◇ Install with pnpm and start now?
Yes
|
◇ Scaffolding project in C:\Users\admin\Desktop\vite-project...
|
◇ Installing dependencies with pnpm...
Downloading @swc/core-win32-x64-msvc@1.15.3: 12.57 MB/12.57 MB, done
Packages: +153
```



# Visit Your New React App

- Open your browser and go to `http://localhost:5173/`

## Get started with React (2/2)

- Press `ctrl + c` to stop the development server.
  - Make sure you click on the terminal window first.
- Open VSCode
- Drag and drop the `vite-project` folder into VSCode
- In the terminal, run:
  - `pnpm run dev`

# Customization (1/2)

- Title
  - Change this line in `index.html`.

```
<title>My very cool title</title>
```

# Customization (2/2)

- Favicon
  - <https://www.favicongenerator.io/>
  - Make favicon and download.
  - Extract `favicon.svg` into `public` folder
  - Change this line in `index.html`.

```
<link rel="icon" type="image/svg+xml" href="/favicon.svg" />
```

# Reset

- Clear content in `src/index.css` and `src/app.css`
- Replace `src/App.jsx` with

```
function App() {  
  return <></>;  
}  
  
export default App;
```

# Styling

- Add the following line at the top of `src/App.jsx`

```
import './App.css';
```

- You can now type CSS in `src/App.css`

# Using CSS

App.jsx

```
import './App.css';
```

# CSS - Example 1

App.css

```
body {  
  font-family: Arial, Helvetica, sans-serif;  
  margin: 1rem 1rem;  
  background-color: #f5f5f5;  
  color: #333333;  
}  
  
h1,  
h2 {  
  color: #2a5d8f;  
}  
  
img {  
  border-radius: 8px;  
}
```



# CSS - Example 2

App.css

```
button {  
  background-color: #2a5d8f;  
  color: white;  
  border: none;  
  padding: 10px 20px;  
  border-radius: 5px;  
  cursor: pointer;  
}  
  
input {  
  padding: 8px;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
  margin-right: 10px;  
}
```

# Dynamic Event

```
<h2>Click Me</h2>  
<button onClick={() => alert("Button clicked!")}>Click me!</button>
```

# State Management

```
import { useState } from "react";

function App() {
  const [count, setCount] = useState(0);

  return (
    <>
      <h2>Count: {count}</h2>
      <button onClick={() => setCount(count + 1)}>Click me!</button>
    </>
  );
}
```

# Send Discord Message

# Get Discord Webhook URL

- Go to your Discord server.
- Go to `Server Settings` > `Integrations` > `Webhooks`.
- Create a new webhook and copy the webhook URL.

## Soundboard

### PEOPLE

Members

Roles

Invites

Access

### APPS

Integrations

App Directory

### MODERATION

Safety Setup

Audit Log

Bans

AutoMod

Enable Community

Server Boost Status

Server template

## Integrations > Webhooks

Webhooks are a simple way to post messages from other apps and websites into Discord using internet mail. [Learn more](#), or try [building one yourself](#).

New Webhook

Posting to #general



Spidey Bot

Created on Aug 6, 2025 by nnnpoo



Minimum Size: 128x128

Name

Spidey Bot

Channel

#general

Copy Webhook URL

Delete Webhook

In `App` return statement

```
<h2>Contact Me</h2>  
<input type="text" id="message" />  
<button onClick={sendWebhook}>Poke Me</button>
```

## In `App` function body

```
// Send discord webhook
function sendWebhook() {
  const webhookUrl = ""; // Replace with your webhook URL

  const message = document.getElementById("message").value || "Hello!";
  const corsProxy = "https://cors.iecmu.com";
  fetch(corsProxy + "/" + webhookUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify({
      content: message,
    }),
  });
}
```



# Build

- `pnpm run build`

# Preview

- `npm run preview`

# Deploy

- Deploy `./dist` folder to Netlify.

# Full Source Code

<https://github.com/inter-workshop-68/react-app>