🟢 Day 15: Deploying Your React App 🚀

Today, you'll **deploy your React application** using **Vercel, Netlify, and GitHub Pages**. You'll also explore **best practices** for production, including performance optimization, error handling, and security measures.

📌 Step 1: Preparing Your React App for Deployment

Before deploying, your app should be **production-ready** to avoid issues.

☑️ 1.1 Check for Errors & Build Production Version

Run the following command to **check for errors** and **generate a production build**:

```
npm run build
```

- This will create a `build/` folder containing optimized files for deployment.
- If there are errors, **fix them before proceeding**.

☑️ 1.2 Optimize Performance

Before deploying, ensure your app is optimized:

- **Remove unnecessary `console.log()`** (use `CTRL + F` in your project).
- **Remove unused dependencies** in `package.json`:

```
npm prune
```

- **Compress large images** (use WebP format for better performance).
- **Enable lazy loading for large components** (explained in Step 4).

☑️ 1.3 Update Metadata (`index.html`)

Your `index.html` (inside `public/`) should contain proper SEO-friendly metadata:

```
<title>My React App</title>
<meta name="description" content="A modern React app" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

✅ This helps search engines index your app properly.

📌 Step 2: Deployment Options

You have **3 options** for deploying your React app:

1️⃣ **Vercel** (Best for React apps, Free & Fast 🚀)
2️⃣ **Netlify** (Great for static sites & JAMstack)
3️⃣ **GitHub Pages** (Free but requires some setup)

◇ Option 1: Deploy with Vercel (Recommended)

1️⃣ Install Vercel CLI (if not installed):

```
npm install -g vercel
```

2️⃣ Login to Vercel:

```
vercel login
```

3️⃣ Deploy your project:

```
vercel
```

✅ Vercel will generate a live **URL** for your app within seconds!

◇ Option 2: Deploy with Netlify

1️⃣ Install Netlify CLI (if not installed):

```
npm install -g netlify-cli
```

2️⃣ Login to Netlify:

```
netlify login
```

3️⃣ Build your project:

```
npm run build
```

4️⃣ Deploy your project:

```
netlify deploy --prod
```

✅ Your app is now live on **Netlify**!

◇ Option 3: Deploy with GitHub Pages

1️⃣ Install GitHub Pages package:

```
npm install gh-pages --save-dev
```

2️⃣ Update `package.json` (Add `homepage` and deploy script):

```
"homepage": "https://your-username.github.io/your-repo",
"scripts": {
  "predeploy": "npm run build",
```

```
  "deploy": "gh-pages -d build"
}
```

3️⃣ **Deploy your app:**

```
npm run deploy
```

✅ Your app will be live at:
`https://your-username.github.io/your-repo`

📌 Step 3: Optimizing for Production

After deploying, follow **best practices** to improve performance.

◇ Enable Lazy Loading (Faster Load Times)

Use **React.lazy()** to load components **only when needed**:
📄 **Lazy loading example**:

```
import React, { Suspense } from "react";

const MovieList = React.lazy(() => import("./MovieList"));

function App() {
  return (
    <Suspense fallback={<p>Loading...</p>}>
      <MovieList />
    </Suspense>
  );
}

export default App;
```

✅ **Faster initial load**
✅ **Loads components only when needed**

◇ Remove Console Logs in Production

To remove `console.log()` from the build, run:

```
npm run build && find build -type f -exec sed -i '' 's/console\.log(.*);//g' {} +
```

✅ Keeps your console **clean** and **professional** in production.

◇ Minify & Optimize Assets

React's `npm run build` automatically **minifies JavaScript and CSS**.
- **Use WebP images** instead of PNG/JPG for better performance.
- **Use gzip compression** (enabled by default in Vercel & Netlify).

📌 Step 4: Error Handling & Security

◇ Use Error Boundaries (Prevent Crashes)

If a React component fails, it shouldn't crash the whole app.

📄 **ErrorBoundary.js**

```
import React from "react";

class ErrorBoundary extends React.Component {
  constructor(props) {
    super(props);
    this.state = { hasError: false };
  }

  static getDerivedStateFromError() {
    return { hasError: true };
  }

  render() {
    if (this.state.hasError) {
      return <h2>Something went wrong!</h2>;
    }
    return this.props.children;
  }
}

export default ErrorBoundary;
```

✅ **Wrap components with `<ErrorBoundary>` to prevent app crashes.**

◇ Secure Your App

✅ **Always use HTTPS** (default in Vercel & Netlify).
✅ **Sanitize user input** to prevent XSS attacks.
✅ **Use `.env` files** to store API keys securely (Don't commit them to GitHub).

📌 Step 5: Final Checklist Before Deployment

✅ Run `npm run build` to check for errors
✅ Remove unnecessary `console.log()`
✅ Optimize images using WebP format
✅ Add error handling (`try...catch` & Error Boundaries)
✅ Test on **mobile devices** and different browsers

📌 Step 6: 🎉 Summary of Day 15

✔ **Deployed React app on Vercel, Netlify, or GitHub Pages**
✔ **Optimized performance with lazy loading & minification**
✔ **Added error handling with Error Boundaries**
✔ **Finalized production-ready version**

🚀 Congratulations! You've completed the 15-day React learning journey! 🎉