Day 1

Here's your 📅 **Day 1 – React Foundation Plan** in your preferred format:

⚙️ React Foundation (Day 1)

🎯 Goal: Master React fundamentals through hands-on coding + mini project

🕒 Duration: ~5 hours

📦 Outcome: You'll be ready to build real apps with React and integrate with APIs from Day 2

✅ Task 1: Setup React App using Visual Studio Template (15 mins)

🧠 Why it matters:

You'll be using the **ASP.NET Core with React** template for all your projects. Knowing where files live is key.

📘 Concepts explained:

- Structure: `ClientApp/`, `Controllers/`, `Data/`, `Startup.cs`
- `npm install` inside `ClientApp`
- Running React + .NET together via proxy

🔨 Example:

```
cd ClientApp
npm install
npm run dev
```

✅ Task 2: Understand JSX, Components, Props (30 mins)

🧠 Why it matters:

Everything in React is a **component**. JSX is the syntax you'll write 95% of the time.

📘 Concepts explained:

- `JSX`: JavaScript + HTML syntax
- `Component`: A function that returns JSX
- `Props`: Input data to components

🔨 Example:

```
function Welcome({ name }) {
  return <h1>Hello, {name}!</h1>;
}
```

✅ Task 3: Learn useState & useEffect (30 mins)

🧠 Why it matters:

They are **core hooks** for managing state and effects. Used in all real projects.

📘 Concepts explained:

- `useState`: Track form input, toggles, etc.
- `useEffect`: Run code after render (like API calls)

🔨 Example:

```
import { useState, useEffect } from 'react';

const Counter = () => {
  const [count, setCount] = useState(0);
  useEffect(() => {
    console.log("Component mounted");
  }, []);
  return <button onClick={() => setCount(count + 1)}>Click {count}</button>;
};
```

✅ Task 4: Add React Router + Navigation (20 mins)

🧠 Why it matters:

Multiple pages? You need routing. Used in all multi-page apps like eCommerce, portals, dashboards.

📘 Concepts explained:

- `BrowserRouter`, `Routes`, `Route`
- Single Page App (SPA) navigation

🔨 Example:

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import About from './pages/About';

<BrowserRouter>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about" element={<About />} />
  </Routes>
</BrowserRouter>
```

✅ Task 5: Build Mini Project – Simple Login Form (2 hrs)

🧠 Why it matters:

This reinforces everything — inputs, state, props, form submission, UI logic.

📘 Concepts explained:

- Controlled form
- Submit event
- Form validation
- useState + conditionals

🔨 Example:

```
function LoginForm() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = e => {
    e.preventDefault();
    alert(`Email: ${email}, Password: ${password}`);
  };

  return (
    <form onSubmit={handleSubmit}>
      <input value={email} onChange={e => setEmail(e.target.value)} placeholder="Email" />
      <input type="password" value={password} onChange={e => setPassword(e.target.value)} placeholder="Password" />
      <button type="submit">Login</button>
    </form>
  );
}
```

✅ Task 6: Push Mini Project to GitHub (30 mins)

🧠 Why it matters:

Version control is essential. Employers expect GitHub history with commits and README.

📘 Concepts explained:

- `git init`, `git add`, `commit`, `push`
- Creating public GitHub repo

🔨 Example:

```
git init
git add .
git commit -m "feat: login form mini project"
git remote add origin https://github.com/your-username/your-repo.git
git push -u origin main
```

✅ By End of Day 1, You Will Have:

Skill / Output	Status
React setup using VS template	✅
JSX, props, components	✅
useState and useEffect basics	✅
React Router navigation	✅
Mini project built and tested	✅
Code pushed to GitHub with commit	✅

Let me know when you're ready to begin **Day 2**, or if you want this plan as a **PDF export**!