Javascript With React Js

Internship Report

On

(java script with react js)

Submitted by

KETAN ANAND

22118128007

BPMCE, MADHEPURA

Bihar, 852128

Submitted To

Mayur Dev Sewak

General Manager, Marketing

& Operations EISystems

Technologies

Student's Declaration

I, **KETAN ANAND** ,a student of **B.Tech** program, Roll No. **22611** of the Department of **3D ANIMATION AND GRAPHICS** In **BPMCE, MADHEPURA, BIHAR, 852128** I do hereby declare that I have completed the mandatory internship in **Eisystems Technologies** under the faculty guideship of **PROF. SUJEET KUMAR** Department of **COMPUTER SCIENCE ENGINEERIG**, **BPMCE, MADHEPURA, 852128**

Ketan Anand

01/06/2025

Endorsements

SIGNATURE

PROF. SUJEET KUMAR

COMPUTER SCIENCE ENGINEERING

BPMCE, MADHEPURA, 852128

SIGNATURE

PROF. MURLIDHI PRASHAD SINGH

COMPUTER SCIENCE ENGINEERING

BPMCE, MADHEPURA, 852128

CERTIFICATE

Internship Completion Certificate

This is to certify that **KETAN ANAND**, a student at **[YourB P MANDAL COLLEGE OF ENGINEERING, MADHEPURA, BIHAR, 852128**, has successfully completed an online internship in **JavaScript with React JS** at **IIT BHU**, organized by **EiSystem**, from **1 May 2025** to **2 june 2025**
During the internship, he actively participated in developing and optimizing front-end user interfaces using React JS, demonstrated strong problem-solving skills, and showed exceptional dedication to learning modern web technologies. Her ability to work collaboratively and adapt to new challenges has been commendable.
We acknowledge her valuable contributions and wish her continued success in all future endeavors.

Date of Issuance: *[Date]*
[Signature]
[Supervisor's Name]
[Designation]
[IIT BHU / EiSystem]

List of Tables

Table of Content

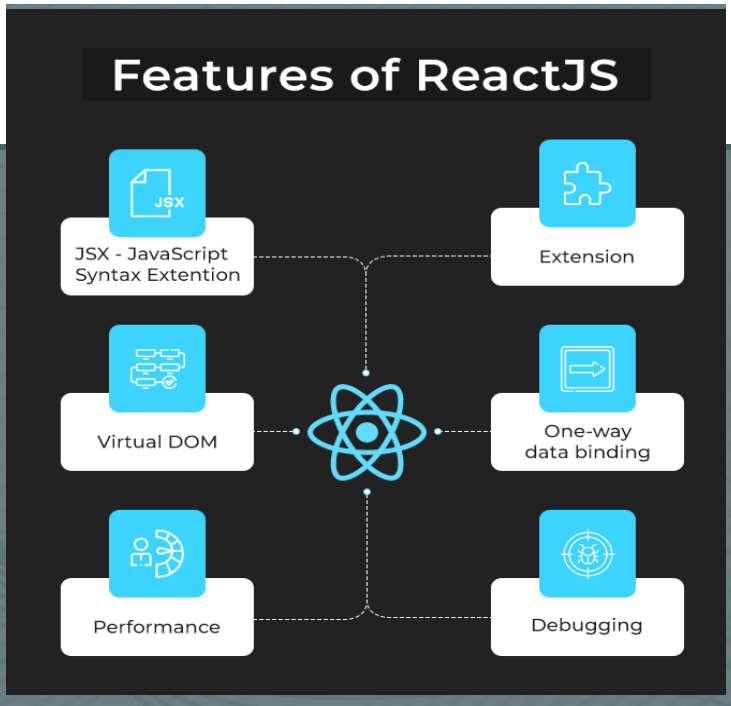
List Of Figures

Nomenclature / Notations

SI.NO.	Notations	Description
1.	JS	JavaScript – a scripting language used to create and control dynamic content.
2.	ES6	ECMAScript 6 – a major update to JavaScript that introduced modern features.
3.	DOM	Document Object Model – interface for accessing and modifying HTML documents.
4.	JSX	JavaScript XML – a syntax extension used in React to describe UI elements.
5.	SPA	Single Page Application – a web app that dynamically updates without reloads.
6.	API	Application Programming Interface – enables data exchange between systems.
7.	useState()	React hook for managing state in functional components.
8.	useEffect()	React hook for side-effects (e.g., API calls, timers) in components.
9.	map()	JavaScript array method used for transforming arrays.
10.	Props	Short for “properties” – data passed from parent to child component in React.
11.	Components	Reusable building blocks in React applications.
12.	Git	Version control system for tracking changes in code.
13.	GitHub	Online platform to host and share code repositories.
14.	async/await	JavaScript syntax for handling asynchronous code execution.
15.	Routing	Mechanism in React (via React Router) to navigate between views/pages.

INTRODUCTION

This project was developed as part of an online internship organized by IIT BHU and EiSystem, focusing on JavaScript with React JS. The main objective was to gain practical experience in modern web development by building a dynamic and interactive website using React. Through this internship, I enhanced my skills in component-based development, user interface design, and the latest JavaScript technologies.



Executive Summary

Learning Objectives

1. **Master the Core Concepts of React JS**
   - Understand component-based architecture, JSX syntax, and the virtual DOM.
2. **Build & Reuse Components**
   - Create both functional and class-based components for modular and maintainable code.
3. **Manage Data Efficiently**
   - Use props and state to control data flow and dynamic behavior in applications.
4. **Leverage React Hooks**
   - Apply hooks like useState, useEffect, and useContext for state and side-effect management.
5. **Create Dynamic User Interfaces**
   - Implement event handling, conditional rendering, and dynamic lists with keys.
6. **Handle Forms with Confidence**
   - Develop forms with validation using controlled and uncontrolled components.
7. **Implement Routing & Navigation**
   - Use React Router to build seamless single-page applications (SPAs).
   -
8. **Integrate APIs & Business Logic**
   - Fetch and display data from external sources within React components.
9. **Style Modern Web Apps**
   - Apply CSS, CSS-in-JS, or frameworks like Bootstrap/Material-UI for responsive design.
10. **Test, Debug & Deploy**
    - Practice best methods for testing, debugging, and deploying React applications.
11. **Enhance Problem-Solving & Optimization Skills**
    - Strengthen analytical abilities and learn strategies for performance optimization.

Summary of Activities Done by the intern

The internship was structured into two main phases: **Training Phase** and **Project Development Phase**. Here's the summary of activities:

Week 1 – JavaScript Fundamentals

- Learned about variables, data types, operators, and functions.
- Practiced conditional statements, loops, and array/object manipulation.
- Completed hands-on coding assignments.

Week 2 – Advanced JavaScript (ES6+)

- Covered topics like arrow functions, destructuring, promises, and async/await.
- Built mini JavaScript programs to practice asynchronous concepts.
- Learned about modular JavaScript and error handling.

Week 3 – React Basics

- Understood JSX syntax, functional components, and props.
- Learned about state and how to handle events in React.
- Created basic components and styled them with CSS or utility frameworks.

Week 4 – React Project Development

- Designed and implemented a **React Portfolio Project**.
- Utilized useState, useEffect, and React Router.
- Integrated a dummy or real API for dynamic content.
- Made the app responsive using Tailwind CSS or Bootstrap.
- Deployed the app (optional) and submitted source code via GitHub.

Overview of the Organization: EISystems Services

Introduction

EISystems Services, commonly known as EISys, is a prominent Indian technology organization with a nationwide presence. Specializing in delivering workshops and training programs, EISys focuses on cutting-edge domains such as Cybersecurity, Machine Learning, Automobiles, Internet of Things (IoT), Robotics, and Social Media. These programs cater to both enterprises and the student community, aiming to bridge the gap between academic knowledge and industry requirements.

Vision, Mission, and Values

👉 **Vision:**

To be a leading provider of technology education, empowering individuals with the skills and knowledge required to excel in the rapidly evolving tech landscape.

👉 **Mission:**

EISystems is committed to delivering high-quality, industry-relevant training programs that foster innovation, practical skills, and a deep understanding of emerging technologies.

👉 **Core Values:**

- Innovation: Embracing new ideas and technologies to stay ahead in the tech education sector.
- Excellence: Maintaining high standards in training delivery and content quality.
- Integrity: Upholding transparency and honesty in all interactions.
- Collaboration: Partnering with academic institutions and industry leaders to provide comprehensive learning experiences.
- Empowerment: Equipping learners with the tools and confidence to succeed in their careers.

Policy in Relation to Intern Roles

EISystems offers structured internship programs designed to provide hands-on experience in various technological domains. The policy emphasizes:

- Eligibility: Open to students from 1st to 4th year of CSE/IT/ICT/MCA/BCA programs.
- Training Delivery: Programs are conducted online with live mentor-led sessions, ensuring interactive and practical learning.
- Certification: Upon successful completion, interns receive a certificate jointly issued by EISystems Services and Technex IIT (BHU) Varanasi.
- Resources Provided: Interns are supplied with study materials, access ID cards, and other necessary resources to facilitate learning.
- Support: Guidance for entrepreneurship or employment support is available upon request.

Project Summary

Idea Behind Making This Project

The main idea was to design a professional and interactive portfolio website using React.js that simulates a real-world business site or service provider portfolio. This project serves not just as a showcase of technical skills, but also as a demonstration of how to build a clean, scalable, and responsive frontend application using modern JavaScript frameworks.

The navigation structure reflects a service-based or personal branding approach — ideal for freelancers, agencies, or startup founders**.**

About the Project

The React Portfolio Website is a Single Page Application (SPA) featuring the following sections:

- **Home –** Clean landing area introducing the user or business.
- **About Us –** Detailed background, values, mission, and expertise.
- **Services –** A dynamic listing of offerings, each with an icon or image.
- **Works –** A portfolio/gallery showcasing completed projects with descriptions.
- **Team –** Introduction to team members with roles and photos.
- **Testimonials –** Customer/client reviews with carousel or cards.
- **Pricing –** Pricing plans or packages for offered services.
- **Blog –** Optional blog posts for updates, learnings, or announcements.
- **Contact –** Form to reach out, along with social links and location.

These sections are smoothly navigable using React Router, and styled with Tailwind CSS (or Bootstrap), ensuring a modern responsive layout across devices.

Software Used in the Project

1. Frontend Framework: React.js
2. Deployment (Optional): GitHub Pages
3. Version Control: Git + GitHub
4. IDE: Visual Studio Code
5. Language: JavaScript (ES6+)

Technical Apparatus Requirements

- Node.js and npm installed on local system.
- Code editor (VS Code preferred).
- Basic knowledge of React hooks, component-based architecture.
- Internet connection for package installation.
- Web browser for testing (Chrome recommended).
- GitHub account (for version control and optionally deployment).

Result / Working of the Project

- Fully responsive website with modern UI.
- SPA structure with fast navigation using React Router.
- Modular component structure for better reusability and scalability.
- Contact form with validation and social media links.
- Clean and minimal animations for smooth UX.
- All sections reflect real-world business functionality.

Research Work Done

Yes, some focused research was conducted:

- UI/UX Research: Studied top portfolio websites for layout and design inspirations.
- React Performance: Learned and applied optimization techniques like React.memo, lazy loading components, etc.
- Deployment Options: Compared GitHub Pages, Netlify, and Vercel; selected based on project size and ease of deployment.
- Form Handling: Researched React form libraries like react-hook-form and validation methods.

Project Planning and Setup

- **Environment Setup**
  - Install Node.js and npm .
  - Use create-react-app to bootstrap the project:

```
npx create-react-app project-name
cd project-name
npm start
```

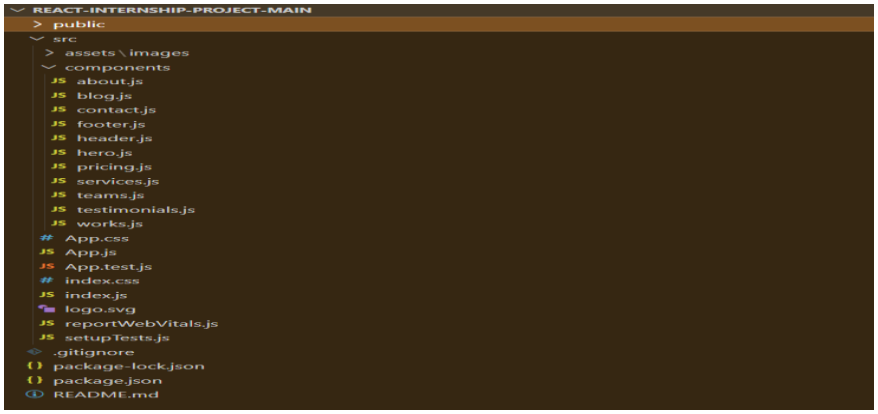
(Fig1)

- **Tools Used**
  - Visual Studio Code (or any preferred code editor).
  - React Developer Tools for Chrome/Firefox.
- **Project Structure**
  - Typical React project structure can be organized by feature or file type:

```
src/
  components/
  pages/
  assets/
  App.js
  index.js
  ...
```

(Fig2)

  - Keep related files (JS, CSS, tests) together for each feature.

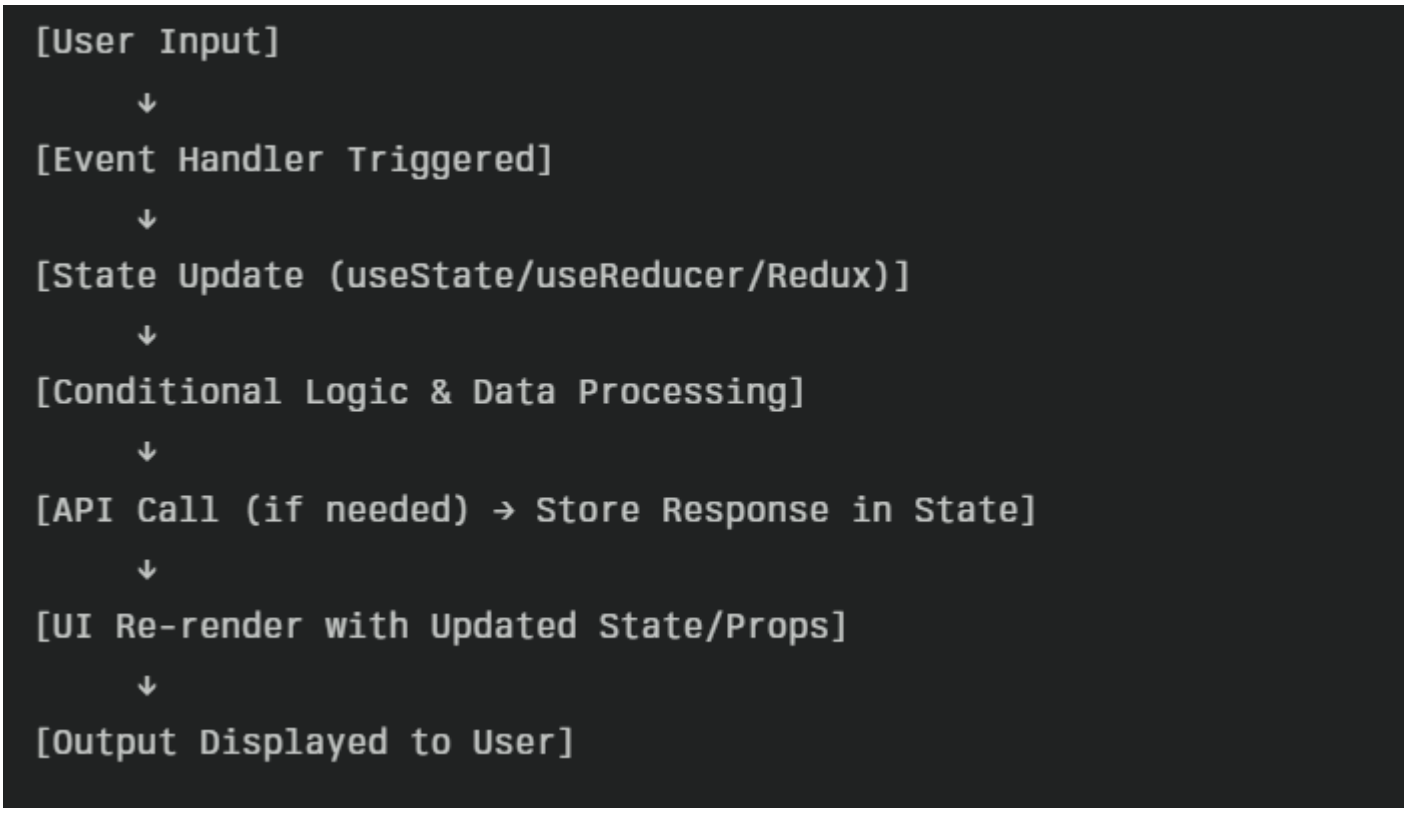


(Fig3)

Process Flow Overview

1. **User Interaction**
    - The user interacts with the UI (e.g., clicks a button, types in a form, or navigates between pages).
2. **Event Handling**
    - React components listen for user events (such as onClick, onChange).
    - Event handler functions are triggered, often updating the component's state.
3. **State Update**
    - The relevant React component uses hooks like useState or useReducer to update its state based on the user action.
    - In larger apps, state might be managed globally using Context API or Redux, where actions are dispatched and reducers update the state accordingly.
4. **Data Flow**
    - Updated state flows down from parent to child components via props.
    - Components re-render automatically when their state or props change, reflecting the latest data in the UI.
5. **Conditional Logic**
    - Components use conditional rendering to display different UI elements or messages based on the current state (e.g., form validation errors, loading spinners, success messages).
6. **API/Data Fetching**
    - On specific events (like form submission or page load), components may fetch data from APIs using fetch or axios within useEffect.
    - Fetched data is stored in state and rendered in the UI.
7. **Output**
    - The UI updates to display the result of the user’s action (e.g., new data, confirmation messages, navigation to another page).

Sample Process Flow Diagram (Textual Representation)



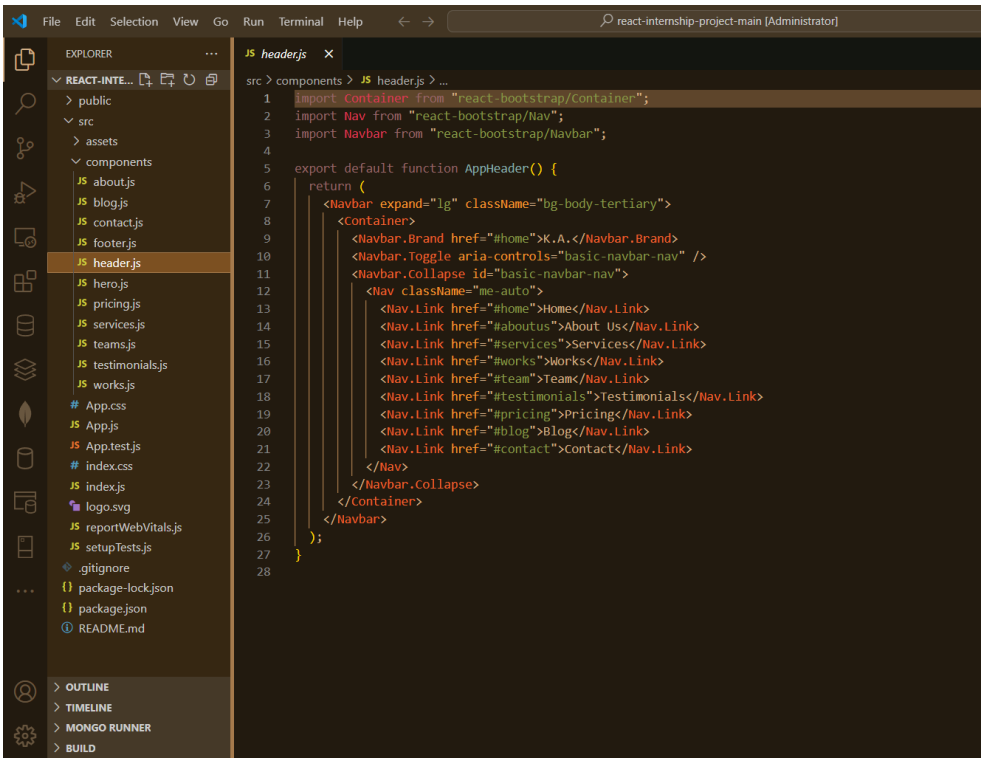
(Fig4)

Code / Program with Supported Screenshots

File Structure Overview

```plaintext
portfolio-react/
├── public/
│   └── index.html
├── src/
│   ├── components/
│   │   ├── Navbar.js
│   │   ├── Footer.js
│   │   ├── Home.js
│   │   ├── About.js
│   │   ├── Services.js
│   │   ├── Works.js
│   │   ├── Team.js
│   │   ├── Testimonials.js
│   │   ├── Pricing.js
│   │   ├── Blog.js
│   │   └── Contact.js
│   ├── App.js
│   ├── index.js
│   └── App.css
```

1. Header.js



2. App.js with Routing Setup

App.js

```
src > App.js > ...
1  import "bootstrap/dist/css/bootstrap.min.css";
2  import "./App.css";
3  import AppHeader from "./components/header";
4  import AppHero from "./components/hero";
5  import AppAbout from "./components/about";
6  import AppServices from "./components/services";
7  import AppWork from "./components/works";
8  import AppTeam from "./components/teams";
9  import AppTestimonials from "./components/testimonials";
10 import AppPricing from "./components/pricing";
11 import AppBlog from "./components/blog";
12 import AppContact from "./components/contact";
13 import AppFooter from "./components/footer";
14 function App() {
15   return (
16     <div className="App">
17       <header id="header">
18         <AppHeader />
19       </header>
20       <main>
21         <AppHero />
22         <AppAbout />
23         <AppServices />
24         <AppWork />
25         <AppTeam />
26         <AppTestimonials />
27         <AppPricing />
28         <AppBlog />
29         <AppContact />
30       </main>
31       <footer id="footer">
32         <AppFooter />
33       </footer>
34     </div>
35   );
36 }
37
38 export default App;
39
```

---

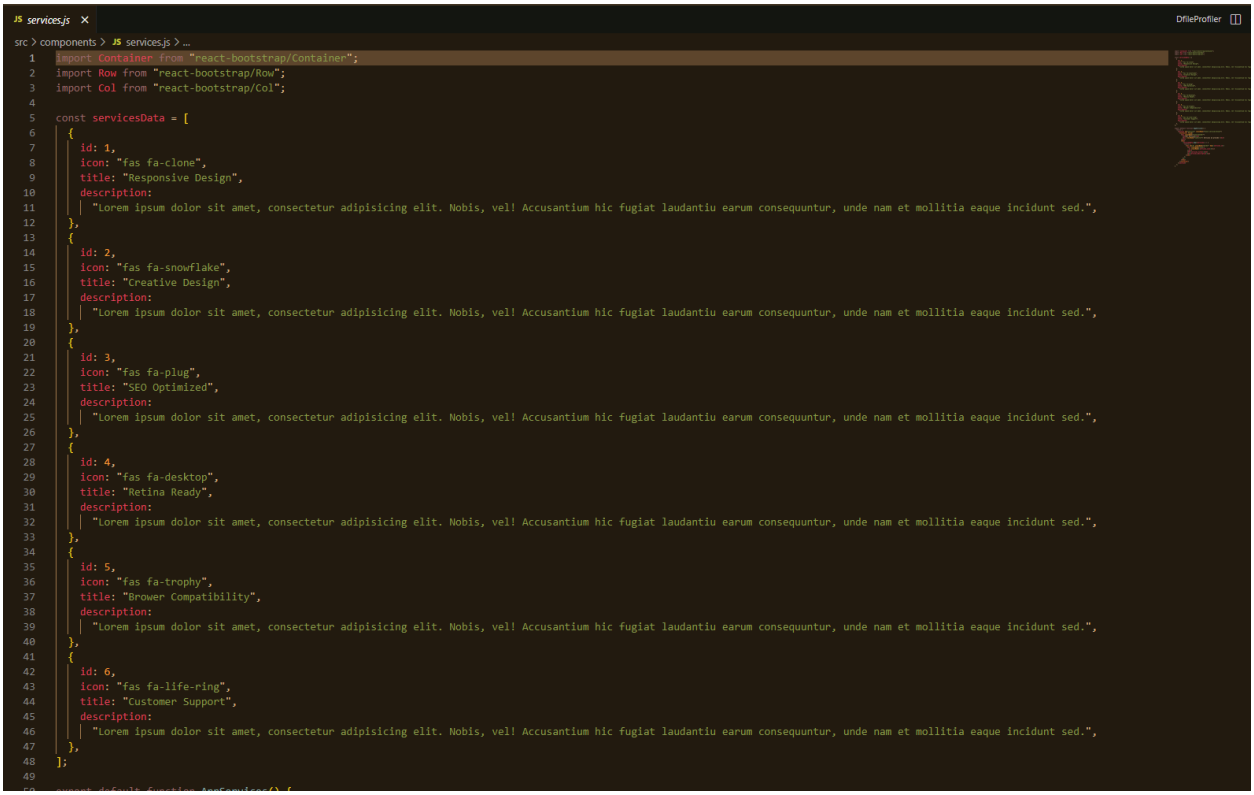
3. Contact.js with Form and useState

contact.js

```
src > components > contact.js > AppContact
1  import Container from "react-bootstrap/Container";
2  import Col from "react-bootstrap/Col";
3  import Form from "react-bootstrap/Form";
4  import Row from "react-bootstrap/Row";
5  import Button from "react-bootstrap/Button";
6
7  export default function AppContact() {
8    return (
9      <section id="contact" className="block contact-block">
10       <Container fluid>
11         <div className="title-holder">
12           <h2>contact us</h2>
13           <div className="subtitle">get connected with us</div>
14         </div>
15         <Form className="contact-form">
16           <Row>
17             <Col sm={4}>
18               <Form.Control
19                 type="text"
20                 placeholder="Enter your full name"
21                 required
22               />
23             </Col>
24             <Col sm={4}>
25               <Form.Control
26                 type="email"
27                 placeholder="Enter your email address"
28                 required
29               />
30             </Col>
31             <Col sm={4}>
32               <Form.Control
33                 type="tel"
34                 placeholder="Enter your contact number"
35                 required
36               />
37             </Col>
38           </Row>
39           <Row>
40             <Col sm={12}>
41               <Form.Control
42                 as="textarea"
43                 placeholder="Enter your message here..."
44                 required
45               />
46             </Col>
47           </Row>
48           <div className="btn-holder">
49             <Button type="submit">Submit</Button>
```

---

4. Sample Services.js with Mapping



```
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";

const servicesData = [
  {
    id: 1,
    icon: "fas fa-clone",
    title: "Responsive Design",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
  {
    id: 2,
    icon: "fas fa-snowflake",
    title: "Creative Design",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
  {
    id: 3,
    icon: "fas fa-plug",
    title: "SEO Optimized",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
  {
    id: 4,
    icon: "fas fa-desktop",
    title: "Retina Ready",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
  {
    id: 5,
    icon: "fas fa-trophy",
    title: "Brower Compatibility",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
  {
    id: 6,
    icon: "fas fa-life-ring",
    title: "Customer Support",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis, vel! Accusantium hic fugiat laudantiu earum consequuntur, unde nam et mollitia eaque incidunt sed.",
  },
];
```

---

5. index.js (React Entry Point)

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Images / Video Links

This section showcases the visual representation and live demonstration of the React.js portfolio project.

Video URL of the Project

Project Demonstration Video (Hosted on video link / Github repositories/ Deployed link):

Hosted URL: https://ketananand76.github.io/react-internship-project/

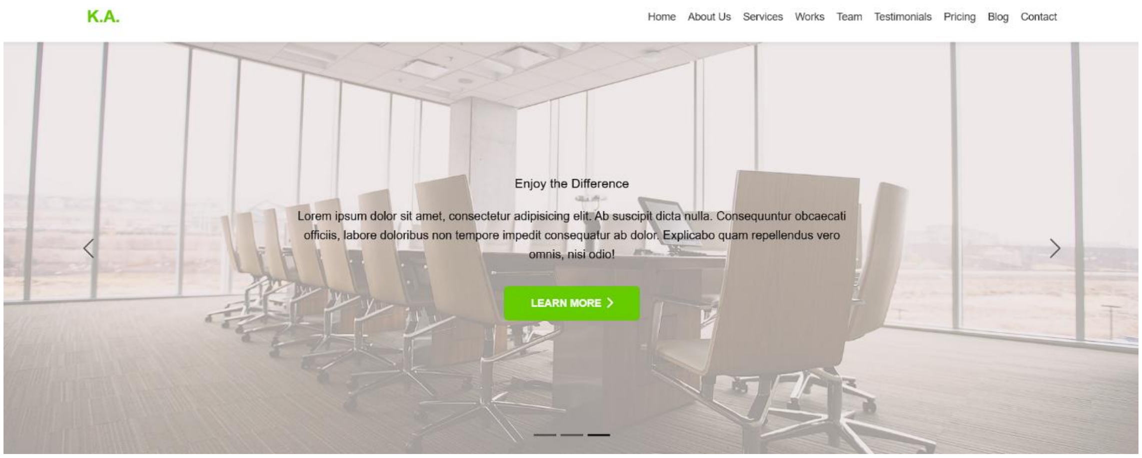
video link: https://drive.google.com/file/d/1LQxPkXXWi92PHe35uLiNlnb9Pih2bLxy/view?usp=drive_link

Github repositories: https://github.com/ketananand76/react-internship-project.git

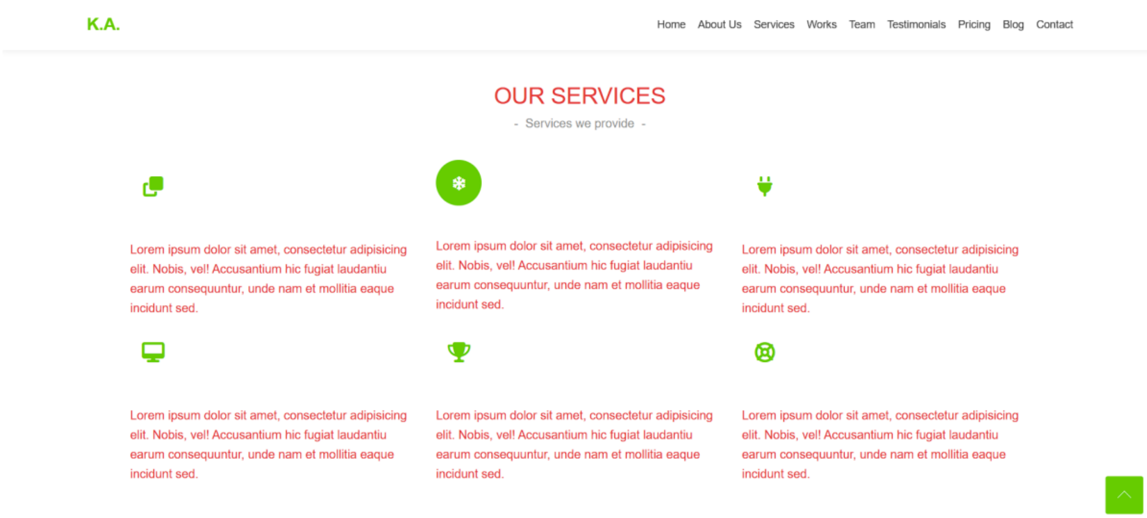
This video provides a complete walkthrough of the website, covering navigation, services, contact form functionality, responsiveness, and animations.

Pictures of the Project

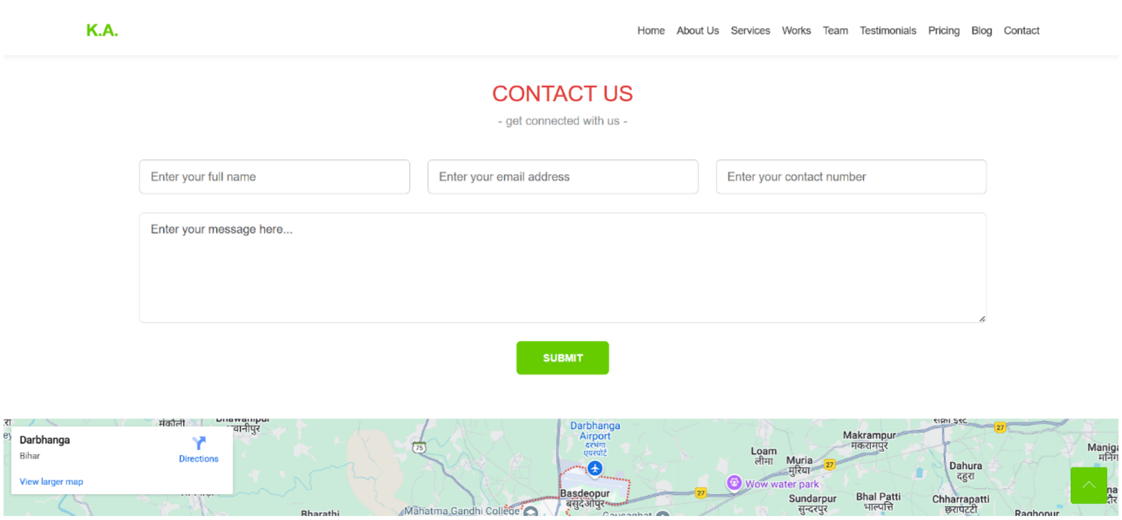
1. **Homepage (Landing Section)**



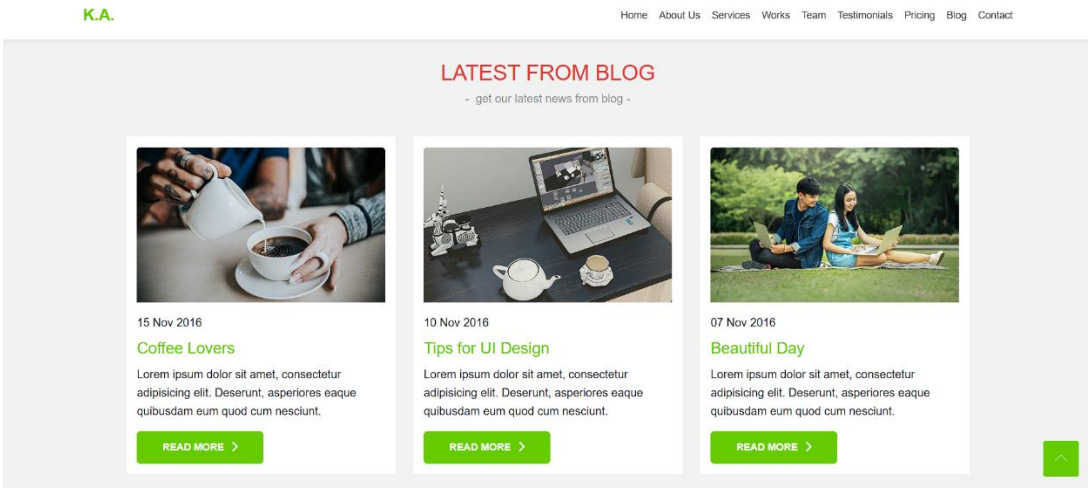
2. **Services Section**



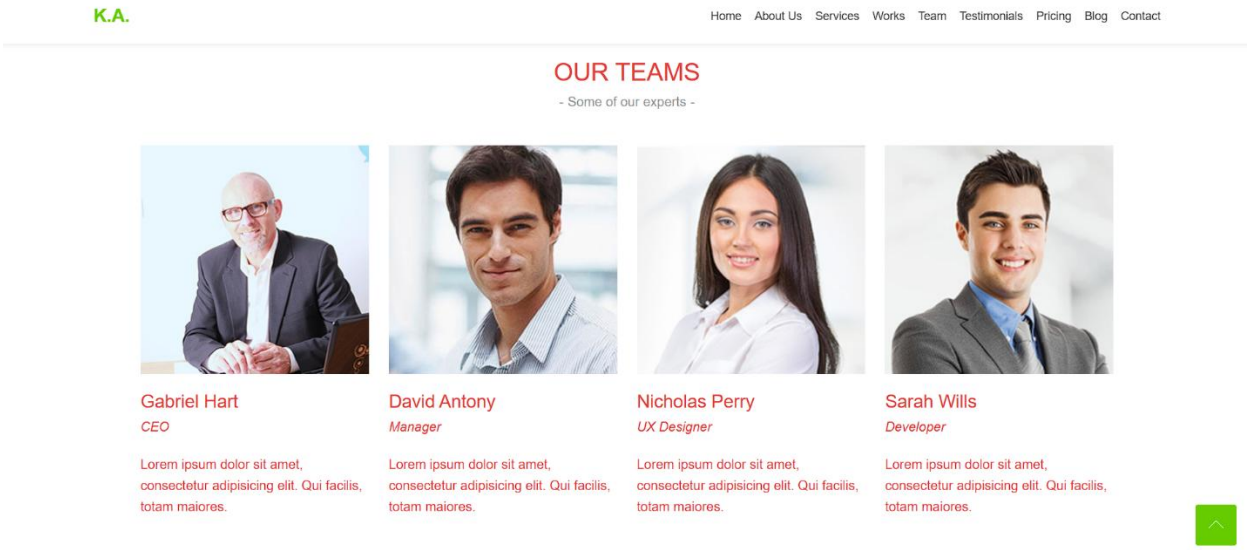
3. **Contact Form**



4. Blog Section



5. Team Section



Student Self-Evaluation of the Short-Term Internship

Evaluation Criteria	Rating
1) Oral Communication	4
2) Written Communication	5
3) Initiative	5
4) Interaction with Staff	4
5) Attitude	5
6) Dependability	5
7) Ability to Learn	5
8) Planning and Organization	4
9) Professionalism	5
10) Creativity	5
11) Quality of Work	5
12) Productivity	5
13) Progress of Learning	5
14) Adaptability to Organization's Culture/ Policies	4
15) OVERALL PERFORMANCE	5

Project Summary:

During this short-term internship, I consistently demonstrated initiative, professionalism, and strong communication skills. I proactively engaged in all training sessions and project development, adapting well to the virtual internship environment. The experience significantly boosted my practical skills in React.js, web development, and teamwork. I am confident in applying these competencies to future academic and professional projects.

Signature of the Student

Annexure 1

Weekly Activity Report

Week No: 4

Week 1	JavaScript Basics and Fundamentals
Week 2	Functions, Loops, Arrays, and Objects
Week 3	Strings, Math, Date, and DOM Basics
Week 4	DOM Advanced + Projects

Annexure 2

Daily Activity Report

Day: (1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/.../28)

Week 1: JavaScript Fundamentals & React Setup

Day	Topics/Activities
1	Introduction to JavaScript, ES6 basics (let/const, arrow functions, template literals)
2	Arrays, objects, functions, destructuring, spread/rest operators
3	JavaScript DOM manipulation basics, events
4	Introduction to React, What is React? Why React?
5	Install Node.js, npm, and VS Code; Create first React app using create-react-app
6	Explore React project structure, run app, understand JSX syntax
7	Build and render your first functional component (Hello World), recap week

Week 2: React Components, Props, State, and Events

Day	Topics/Activities
8	Functional vs Class components, create multiple components
9	Passing data with props, props vs state
10	Using useState for local state management
11	Handling events (onClick, onChange), binding event handlers
12	Conditional rendering (if/else, ternary, &&)
13	Rendering lists with .map(), using keys
14	Build a small project: Simple To-Do List or Counter app

Week 3: Forms, Hooks, Lifecycle, and Styling

Day	Topics/Activities
15	Controlled vs uncontrolled components, handling forms
16	Form validation, handling form submission
17	Introduction to React Hooks: useEffect, side effects
18	More on hooks: useContext, custom hooks (basics)
19	Component lifecycle (mount, update, unmount)
20	Styling in React: CSS, inline styles, CSS Modules, styled-components
21	Build a small project: Contact Form or Feedback Form

Week 4: Routing, API Integration, Final Project & Deployment

Day	Topics/Activities
22	Introduction to React Router, setting up routes
23	Navigation, route parameters, nested routes
24	Fetching data from APIs using fetch or axios
25	Displaying API data, handling loading and error states
26	Project planning: Choose and design a final project (e.g., Blog, Movie List, Weather App)
27	Work on final project: Structure, components, data flow
28	Complete final project, deploy to Netlify/Vercel/GitHub Pages, prepare project report