**React Projects Assignment**

**1. myfirstreact Project**

**Code in App.js:**

import React from 'react';

function App() {

return (

<div>

<h1>Welcome to the first session of React</h1>

</div>

);

}

export default App;

**Output:** Welcome to the first session of React (as a heading on the browser page)

**2. StudentApp Project**

**Code in Home.js:**

import React from 'react';

class Home extends React.Component {

render() {

return <h2>Welcome to the Home page of Student Management Portal</h2>;

}

}

export default Home;

**Code in About.js:**

import React from 'react';

class About extends React.Component {

render() {

return <h2>Welcome to the About page of the Student Management Portal</h2>;

}

}

export default About;

**Code in Contact.js:**

import React from 'react';

class Contact extends React.Component {

render() {

return <h2>Welcome to the Contact page of the Student Management Portal</h2>;

}

}

export default Contact;

**Code in App.js:**

import React from 'react';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

return (

<div>

<Home />

<About />

<Contact />

</div>

);

}

export default App;

**Output:** Displays 3 headings for Home, About, and Contact pages respectively.

**3. scorecalculatorapp Project**

**Code in CalculateScore.js:**

import React from 'react';

import '../Stylesheets/mystyle.css';

function CalculateScore(props) {

const average = props.total / props.goal;

return (

<div className="score-box">

<h2>{props.name} - {props.school}</h2>

<p>Average Score: {average}</p>

</div>

);

}

export default CalculateScore;

**CSS in mystyle.css:**

.score-box {

border: 1px solid black;

padding: 20px;

margin: 10px;

background-color: #f5f5f5;

}

**Code in App.js:**

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div>

<CalculateScore name="John" school="Greenwood High" total={450} goal={5} />

</div>

);

}

export default App;

**Output:** Displays student name, school and calculated average score in styled box.

**4. blogapp Project**

**Code in Post.js:**

import React from 'react';

function Post({ title, body }) {

return (

<div>

<h2>{title}</h2>

<p>{body}</p>

</div>

);

}

export default Post;

**Code in Posts.js:**

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = { posts: [], hasError: false };

}

loadPosts = () => {

fetch('https://jsonplaceholder.typicode.com/posts')

.then(res => res.json())

.then(data => this.setState({ posts: data }))

.catch(error => this.setState({ hasError: true }));

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch() {

alert('An error occurred while loading posts');

}

render() {

return (

<div>

{this.state.posts.map(post => (

<Post key={post.id} title={post.title} body={post.body} />

))}

</div>

);

}

}

export default Posts;

**Code in App.js:**

import React from 'react';

import Posts from './Posts';

function App() {

return <Posts />;

}

export default App;

**Output:** Dynamically lists post titles and content fetched from API.

**5. counterapp Project**

**Code in Counter.js:**

import React, { useState } from 'react';

import '../Stylesheets/counter.css';

function Counter() {

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

const increment = () => setCount(count + 1);

const decrement = () => setCount(count - 1);

const reset = () => setCount(0);

return (

<div className="counter-container">

<h2>Counter Application</h2>

<div className="count-display">

<h3>Count: {count}</h3>

</div>

<div className="button-group">

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

<button onClick={reset}>Reset</button>

</div>

</div>

);

}

export default Counter;

**CSS in counter.css:**

.counter-container {

text-align: center;

margin: 50px;

padding: 20px;

border: 2px solid #007bff;

border-radius: 10px;

}

.count-display {

font-size: 24px;

color: #007bff;

margin: 20px 0;

}

.button-group button {

margin: 0 10px;

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

border: none;

border-radius: 5px;

background-color: #007bff;

color: white;

}

.button-group button:hover {

background-color: #0056b3;

}

**Code in App.js:**

import React from 'react';

import Counter from './Components/Counter';

function App() {

return (

<div>

<Counter />

</div>

);

}

export default App;

**Output:** Displays a counter with increment, decrement, and reset buttons. Shows current count value and updates dynamically when buttons are clicked.

**6. todoapp Project**

**Code in TodoItem.js:**

import React from 'react';

function TodoItem({ todo, onToggle, onDelete }) {

return (

<div className="todo-item">

<input

type="checkbox"

checked={todo.completed}

onChange={() => onToggle(todo.id)}

/>

<span className={todo.completed ? 'completed' : ''}>

{todo.text}

</span>

<button onClick={() => onDelete(todo.id)}>Delete</button>

</div>

);

}

export default TodoItem;

**Code in TodoList.js:**

import React, { useState } from 'react';

import TodoItem from './TodoItem';

import '../Stylesheets/todo.css';

function TodoList() {

const [todos, setTodos] = useState([]);

const [inputValue, setInputValue] = useState('');

const addTodo = () => {

if (inputValue.trim()) {

const newTodo = {

id: Date.now(),

text: inputValue,

completed: false

};

setTodos([...todos, newTodo]);

setInputValue('');

}

};

const toggleTodo = (id) => {

setTodos(todos.map(todo =>

todo.id === id ? { ...todo, completed: !todo.completed } : todo

));

};

const deleteTodo = (id) => {

setTodos(todos.filter(todo => todo.id !== id));

};

return (

<div className="todo-container">

<h2>Todo List Application</h2>

<div className="input-section">

<input

type="text"

value={inputValue}

onChange={(e) => setInputValue(e.target.value)}

placeholder="Enter a task"

/>

<button onClick={addTodo}>Add Task</button>

</div>

<div className="todo-list">

{todos.map(todo => (

<TodoItem

key={todo.id}

todo={todo}

onToggle={toggleTodo}

onDelete={deleteTodo}

/>

))}

</div>

</div>

);

}

export default TodoList;

**CSS in todo.css:**

.todo-container {

max-width: 500px;

margin: 50px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 10px;

}

.input-section {

display: flex;

margin-bottom: 20px;

}

.input-section input {

flex: 1;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

.input-section button {

padding: 10px 20px;

margin-left: 10px;

background-color: #28a745;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.todo-item {

display: flex;

align-items: center;

padding: 10px;

border-bottom: 1px solid #eee;

}

.completed {

text-decoration: line-through;

color: #888;

}

**Code in App.js:**

import React from 'react';

import TodoList from './Components/TodoList';

function App() {

return (

<div>

<TodoList />

</div>

);

}

export default App;

**Output:** Displays a todo list application with input field to add tasks, checkboxes to mark completion, and delete buttons. Tasks can be added, marked as complete, and removed dynamically.

**7. weatherapp Project**

**Code in WeatherCard.js:**

import React from 'react';

function WeatherCard({ weather }) {

if (!weather) {

return <div>Loading weather data...</div>;

}

return (

<div className="weather-card">

<h3>{weather.city}</h3>

<div className="temperature">

<span>{weather.temperature}°C</span>

</div>

<div className="description">

<p>{weather.description}</p>

</div>

<div className="details">

<p>Humidity: {weather.humidity}%</p>

<p>Wind Speed: {weather.windSpeed} km/h</p>

</div>

</div>

);

}

export default WeatherCard;

**Code in Weather.js:**

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

import WeatherCard from './WeatherCard';

import '../Stylesheets/weather.css';

function Weather() {

const [weather, setWeather] = useState(null);

const [city, setCity] = useState('Delhi');

const mockWeatherData = {

Delhi: {

city: 'Delhi',

temperature: 32,

description: 'Sunny',

humidity: 45,

windSpeed: 15

},

Mumbai: {

city: 'Mumbai',

temperature: 28,

description: 'Partly Cloudy',

humidity: 70,

windSpeed: 12

},

Bangalore: {

city: 'Bangalore',

temperature: 25,

description: 'Pleasant',

humidity: 60,

windSpeed: 8

}

};

useEffect(() => {

// Simulate API call

setTimeout(() => {

setWeather(mockWeatherData[city]);

}, 1000);

}, [city]);

const handleCityChange = (newCity) => {

setCity(newCity);

setWeather(null);

};

return (

<div className="weather-container">

<h2>Weather Application</h2>

<div className="city-selector">

<button onClick={() => handleCityChange('Delhi')}>Delhi</button>

<button onClick={() => handleCityChange('Mumbai')}>Mumbai</button>

<button onClick={() => handleCityChange('Bangalore')}>Bangalore</button>

</div>

<WeatherCard weather={weather} />

</div>

);

}

export default Weather;

**CSS in weather.css:**

.weather-container {

max-width: 400px;

margin: 50px auto;

text-align: center;

}

.city-selector {

margin: 20px 0;

}

.city-selector button {

margin: 0 5px;

padding: 10px 15px;

background-color: #17a2b8;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.weather-card {

background: linear-gradient(135deg, #74b9ff, #0984e3);

color: white;

padding: 30px;

border-radius: 15px;

box-shadow: 0 4px 6px rgba(0,0,0,0.1);

}

.temperature span {

font-size: 48px;

font-weight: bold;

}

**Code in App.js:**

import React from 'react';

import Weather from './Components/Weather';

function App() {

return (

<div>

<Weather />

</div>

);

}

export default App;

**Output:** Displays a weather application showing temperature, description, humidity, and wind speed for selected cities. Users can switch between cities using buttons.

**8. userprofileapp Project**

**Code in ProfileCard.js:**

import React from 'react';

function ProfileCard({ user }) {

return (

<div className="profile-card">

<img src={user.avatar} alt={`${user.name} avatar`} className="avatar" />

<h3>{user.name}</h3>

<p className="email">{user.email}</p>

<p className="bio">{user.bio}</p>

<div className="skills">

<h4>Skills:</h4>

<div className="skill-tags">

{user.skills.map((skill, index) => (

<span key={index} className="skill-tag">{skill}</span>

))}

</div>

</div>

</div>

);

}

export default ProfileCard;

**Code in UserProfile.js:**

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

import ProfileCard from './ProfileCard';

import '../Stylesheets/profile.css';

function UserProfile() {

const [users, setUsers] = useState([]);

const mockUsers = [

{

id: 1,

name: 'Alice Johnson',

email: 'alice@example.com',

bio: 'Frontend Developer passionate about React and UI/UX design.',

avatar: 'https://via.placeholder.com/150/FF6B6B/FFFFFF?text=AJ',

skills: ['React', 'JavaScript', 'CSS', 'HTML']

},

{

id: 2,

name: 'Bob Smith',

email: 'bob@example.com',

bio: 'Full-stack developer with expertise in Node.js and databases.',

avatar: 'https://via.placeholder.com/150/4ECDC4/FFFFFF?text=BS',

skills: ['Node.js', 'MongoDB', 'Express', 'React']

},

{

id: 3,

name: 'Carol Davis',

email: 'carol@example.com',

bio: 'Data scientist specializing in machine learning and analytics.',

avatar: 'https://via.placeholder.com/150/45B7D1/FFFFFF?text=CD',

skills: ['Python', 'TensorFlow', 'SQL', 'R']

}

];

useEffect(() => {

setUsers(mockUsers);

}, []);

return (

<div className="user-profile-container">

<h2>User Profiles</h2>

<div className="profiles-grid">

{users.map(user => (

<ProfileCard key={user.id} user={user} />

))}

</div>

</div>

);

}

export default UserProfile;

**CSS in profile.css:**

.user-profile-container {

max-width: 1200px;

margin: 50px auto;

padding: 20px;

}

.profiles-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 20px;

margin-top: 20px;

}

.profile-card {

background: white;

border-radius: 10px;

padding: 20px;

box-shadow: 0 4px 6px rgba(0,0,0,0.1);

text-align: center;

}

.avatar {

width: 100px;

height: 100px;

border-radius: 50%;

margin-bottom: 15px;

}

.email {

color: #666;

margin-bottom: 15px;

}

.bio {

font-style: italic;

margin-bottom: 20px;

}

.skill-tags {

display: flex;

flex-wrap: wrap;

gap: 8px;

justify-content: center;

}

.skill-tag {

background-color: #007bff;

color: white;

padding: 4px 8px;

border-radius: 12px;

font-size: 12px;

}

**Code in App.js:**

import React from 'react';

import UserProfile from './Components/UserProfile';

function App() {

return (

<div>

<UserProfile />

</div>

);

}

export default App;

**Output:** Displays user profile cards in a grid layout showing avatar, name, email, bio, and skills for multiple users. Each profile is styled as an attractive card component.

**9. quizapp Project**

**Code in Question.js:**

import React from 'react';

function Question({ question, onAnswer, selectedAnswer }) {

return (

<div className="question-container">

<h3>{question.text}</h3>

<div className="options">

{question.options.map((option, index) => (

<button

key={index}

className={`option-btn ${selectedAnswer === option ? 'selected' : ''}`}

onClick={() => onAnswer(option)}

>

{option}

</button>

))}

</div>

</div>

);

}

export default Question;

**Code in Quiz.js:**

import React, { useState } from 'react';

import Question from './Question';

import '../Stylesheets/quiz.css';

function Quiz() {

const questions = [

{

id: 1,

text: 'What is the capital of France?',

options: ['London', 'Berlin', 'Paris', 'Madrid'],

correct: 'Paris'

},

{

id: 2,

text: 'Which planet is known as the Red Planet?',

options: ['Venus', 'Mars', 'Jupiter', 'Saturn'],

correct: 'Mars'

},

{

id: 3,

text: 'What is 2 + 2?',

options: ['3', '4', '5', '6'],

correct: '4'

}

];

const [currentQuestion, setCurrentQuestion] = useState(0);

const [selectedAnswer, setSelectedAnswer] = useState('');

const [score, setScore] = useState(0);

const [showResult, setShowResult] = useState(false);

const handleAnswer = (answer) => {

setSelectedAnswer(answer);

};

const nextQuestion = () => {

if (selectedAnswer === questions[currentQuestion].correct) {

setScore(score + 1);

}

if (currentQuestion + 1 < questions.length) {

setCurrentQuestion(currentQuestion + 1);

setSelectedAnswer('');

} else {

setShowResult(true);

}

};

const resetQuiz = () => {

setCurrentQuestion(0);

setSelectedAnswer('');

setScore(0);

setShowResult(false);

};

if (showResult) {

return (

<div className="quiz-container">

<div className="result">

<h2>Quiz Complete!</h2>

<p>Your Score: {score} out of {questions.length}</p>

<button onClick={resetQuiz}>Restart Quiz</button>

</div>

</div>

);

}

return (

<div className="quiz-container">

<h2>React Quiz App</h2>

<div className="progress">

Question {currentQuestion + 1} of {questions.length}

</div>

<Question

question={questions[currentQuestion]}

onAnswer={handleAnswer}

selectedAnswer={selectedAnswer}

/>

<button

className="next-btn"

onClick={nextQuestion}

disabled={!selectedAnswer}

>

{currentQuestion + 1 === questions.length ? 'Finish Quiz' : 'Next Question'}

</button>

</div>

);

}

export default Quiz;

**CSS in quiz.css:**

.quiz-container {

max-width: 600px;

margin: 50px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 10px;

text-align: center;

}

.progress {

background-color: #f8f9fa;

padding: 10px;

border-radius: 5px;

margin-bottom: 20px;

}

.question-container {

margin-bottom: 30px;

}

.options {

display: grid;

gap: 10px;

margin-top: 20px;

}

.option-btn {

padding: 15px;

border: 2px solid #ddd;

background-color: white;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

}

.option-btn:hover {

background-color: #f8f9fa;

}

.option-btn.selected {

background-color: #007bff;

color: white;

border-color: #007bff;

}

.next-btn {

padding: 15px 30px;

background-color: #28a745;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.next-btn:disabled {

background-color: #6c757d;

cursor: not-allowed;

}

.result {

padding: 30px;

}

**Code in App.js:**

import React from 'react';

import Quiz from './Components/Quiz';

function App() {

return (

<div>

<Quiz />

</div>

);

}

export default App;

**Output:** Displays an interactive quiz application with multiple choice questions. Shows progress, allows answer selection, calculates score, and displays final results with option to restart.

**10. galleryapp Project**

**Code in ImageCard.js:**

import React from 'react';

function ImageCard({ image, onClick }) {

return (

<div className="image-card" onClick={() => onClick(image)}>

<img src={image.thumbnail} alt={image.title} />

<div className="image-info">

<h4>{image.title}</h4>

<p>{image.description}</p>

</div>

</div>

);

}

export default ImageCard;

**Code in Modal.js:**

import React from 'react';

function Modal({ image, onClose }) {

if (!image) return null;

return (

<div className="modal-overlay" onClick={onClose}>

<div className="modal-content" onClick={(e) => e.stopPropagation()}>

<button className="close-btn" onClick={onClose}>×</button>

<img src={image.fullSize} alt={image.title} />

<div className="modal-info">

<h3>{image.title}</h3>

<p>{image.description}</p>

<small>By {image.author}</small>

</div>

</div>

</div>

);

}

export default Modal;

**Code in Gallery.js:**

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

import ImageCard from './ImageCard';

import Modal from './Modal';

import '../Stylesheets/gallery.css';

function Gallery() {

const [images, setImages] = useState([]);

const [selectedImage, setSelectedImage] = useState(null);

const mockImages = [

{

id: 1,

title: 'Mountain Landscape',

description: 'Beautiful mountain view at sunset',

author: 'John Doe',

thumbnail: 'https://via.placeholder.com/300x200/FF6B6B/FFFFFF?text=Mountain',

fullSize: 'https://via.placeholder.com/800x600/FF6B6B/FFFFFF?text=Mountain+HD'

},

{

id: 2,

title: 'Ocean Waves',

description: 'Peaceful ocean waves on a sunny day',

author: 'Jane Smith',

thumbnail: 'https://via.placeholder.com/300x200/4ECDC4/FFFFFF?text=Ocean',

fullSize: 'https://via.placeholder.com/800x600/4ECDC4/FFFFFF?text=Ocean+HD'

},

{

id: 3,

title: 'City Skyline',

description: 'Modern city skyline at night',

author: 'Mike Johnson',

thumbnail: 'https://via.placeholder.com/300x200/45B7D1/FFFFFF?text=City',

fullSize: 'https://via.placeholder.com/800x600/45B7D1/FFFFFF?text=City+HD'

},

{

id: 4,

title: 'Forest Path',

description: 'Serene path through green forest',

author: 'Sarah Wilson',

thumbnail: 'https://via.placeholder.com/300x200/96CEB4/FFFFFF?text=Forest',

fullSize: 'https://via.placeholder.com/800x600/96CEB4/FFFFFF?text=Forest+HD'

},

{

id: 5,

title: 'Desert Dunes',

description: 'Golden sand dunes in the desert',

author: 'Alex Brown',

thumbnail: 'https://via.placeholder.com/300x200/FFEAA7/000000?text=Desert',

fullSize: 'https://via.placeholder.com/800x600/FFEAA7/000000?text=Desert+HD'

},

{

id: 6,

title: 'Snowy Mountains',

description: 'Snow-covered mountain peaks',

author: 'Emma Davis',

thumbnail: 'https://via.placeholder.com/300x200/DDA0DD/FFFFFF?text=Snow',

fullSize: 'https://via.placeholder.com/800x600/DDA0DD/FFFFFF?text=Snow+HD'

}

];

useEffect(() => {

setImages(mockImages);

}, []);

const openModal = (image) => {

setSelectedImage(image);

};

const closeModal = () => {

setSelectedImage(null);

};

return (

<div className="gallery-container">

<h2>Photo Gallery</h2>

<div className="gallery-grid">

{images.map(image => (

<ImageCard

key={image.id}

image={image}

onClick={openModal}

/>

))}

</div>

<Modal image={selectedImage} onClose={closeModal} />

</div>

);

}

export default Gallery;

**CSS in gallery.css:**

.gallery-container {

max-width: 1200px;

margin: 50px auto;

padding: 20px;

}

.gallery-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));

gap: 20px;

margin-top: 20px;

}

.image-card {

background: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 4px 6px rgba(0,0,0,0.1);

cursor: pointer;

transition: transform 0.3s ease;

}

.image-card:hover {

transform: translateY(-5px);

}

.image-card img {

width: 100%;

height: 200px;

object-fit: cover;

}

.image-info {

padding: 15px;

}

.image-info h4 {

margin: 0 0 10px 0;

color: #333;

}

.image-info p {

margin: 0;

color: #666;

font-size: 14px;

}

.modal-overlay {

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: rgba(0,0,0,0.8);

display: flex;

align-items: center;

justify-content: center;

z-index: 1000;

}

.modal-content {

background: white;

border-radius: 10px;

max-width: 90vw;

max-height: 90vh;

position: relative;

overflow: auto;

}

.close-btn {

position: absolute;

top: 10px;

right: 15px;

background: none;

border: none;

font-size: 30px;

color: white;

cursor: pointer;

z-index: 1001;

}

.modal-content img {

width: 100%;

height: auto;

display: block;

}

.modal-info {

padding: 20px;

}

.modal-info h3 {

margin: 0 0 10px 0;

color: #333;

}

.modal-info p {

margin: 0 0 10px 0;

color: #666;

}

.modal-info small {

color: #999;

font-style: italic;

}

**Code in App.js:**

import React from 'react';

import Gallery from './Components/Gallery';

function App() {

return (

<div>

<Gallery />

</div>

);

}

export default App;

**Output:** Displays a responsive photo gallery with thumbnail images arranged in a grid layout. Clicking on any image opens a modal popup showing the full-size image with title, description, and author information. Users can close the modal by clicking the close button or clicking outside the modal content.