Week 7
React

Below are **copy-paste-ready minimal implementations** for **Labs 9–13**. Each lab is its own React app; assume you start with:

```
npx create-react-app <appname>
cd <appname>
```

Then replace/add the shown files.

Lab 9: ES6 Features in React (`cricketapp`)

Goals: `ListofPlayers`, filtering with arrow functions, destructuring, merging arrays, conditional display via flag.

src/components/ListofPlayers.js

```
import React from 'react';

function ListofPlayers({ players, flag }) {
  // Filter players with score < 70
  const lowScorers = players.filter(p => p.score < 70);

  return (
    <div>
      <h2>All Players</h2>
      <ul>
        {players.map(p => (
          <li key={p.name}>
            {p.name} - {p.score}
          </li>
        ))}
      </ul>

      <h3>Players with score below 70</h3>
      <ul>
        {lowScorers.map(p => (
          <li key={p.name}>
            {p.name} - {p.score}
          </li>
        ))}
      </ul>
    </div>
```

```
  );
}

export default ListofPlayers;
```

src/components/IndianPlayers.js

```
import React from 'react';

function IndianPlayers() {
  const T20players = ['Virat', 'Rohit', 'Hardik'];
  const RanjiTrophy = ['PlayerA', 'PlayerB'];

  // Merge arrays using spread
  const merged = [...T20players, ...RanjiTrophy];

  // Destructure odd/even team example
  const [odd1, even1, odd2, even2, ...rest] = merged;

  return (
    <div>
      <h2>Indian Players</h2>
      <p>Merged List: {merged.join(', ')}</p>
      <p>Destructured (example):</p>
      <ul>
        <li>odd1: {odd1}</li>
        <li>even1: {even1}</li>
        <li>odd2: {odd2}</li>
        <li>even2: {even2}</li>
        {rest.length > 0 && <li>rest: {rest.join(', ')}</li>}
      </ul>
    </div>
  );
}

export default IndianPlayers;
```

src/App.js

```
import React from 'react';
import ListofPlayers from './components/ListofPlayers';
import IndianPlayers from './components/IndianPlayers';

function App() {
```

Week 7

```
  const flag = true; // toggle to false to switch display
logic

  const players = [
    { name: 'Player1', score: 85 },
    { name: 'Player2', score: 65 },
    { name: 'Player3', score: 72 },
    { name: 'Player4', score: 50 },
    { name: 'Player5', score: 90 },
    { name: 'Player6', score: 68 },
    { name: 'Player7', score: 77 },
    { name: 'Player8', score: 55 },
    { name: 'Player9', score: 80 },
    { name: 'Player10', score: 66 },
    { name: 'Player11', score: 95 },
  ];

  return (
    <div style={{ padding: '20px' }}>
      <h1>Cricket App</h1>
      {flag ? (
        <>
          <ListofPlayers players={players} flag={flag} />
          <IndianPlayers />
        </>
      ) : (
        <div>
          <p>Flag is false. Alternative view can go here.</p>
          <IndianPlayers />
        </div>
      )}
    </div>
  );
}

export default App;
```

Week 7

Lab 10: JSX & Inline CSS (`officespacerentalapp`)

Use JSX, loop through office spaces, conditional inline styling for rent.

src/App.js

```
import React from 'react';

function App() {
  const offices = [
    { name: 'Downtown Suite', rent: 55000, address: '123 Main
St' },
    { name: 'Riverside Office', rent: 75000, address: '456
River Rd' },
    { name: 'Tech Hub', rent: 60000, address: '789 Silicon
Ave' },
  ];

  return (
    <div style={{ padding: '20px', fontFamily: 'sans-
serif' }}>
      <h1>Office Space Rental</h1>
      <div>
        {offices.map((office, idx) => (
          <div
            key={idx}
            style={{
              border: '1px solid #ccc',
              marginBottom: '15px',
              padding: '10px',
              borderRadius: '8px',
            }}
          >
            <h2>{office.name}</h2>
            <p>
              <strong>Address:</strong> {office.address}
            </p>
            <p>
              <strong>Rent:</strong>{' '}
              <span
                style={{
                  color: office.rent < 60000 ? 'red' :
'green',
                  fontWeight: 'bold',
```

Week 7

```
                }}
              >
                ₹{office.rent.toLocaleString()}
              </span>
            </p>
            <img
              alt="office"
              src="https://via.placeholder.com/300x150?
text=Office+Image"
              style={{ width: '300px', borderRadius: '5px' }}
            />
          </div>
        ))}
      </div>
    </div>
  );
}

export default App;
```

Week 7

Lab 11: Event Handling (`eventexamplesapp`)

Counter with multiple handlers, argument passing, synthetic event, currency converter.

src/components/Counter.js

```
import React, { Component } from 'react';

class Counter extends Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  sayHello = () => {
    console.log('Hello! Incrementing...');
  };

  increment = () => {
    this.sayHello();
    this.setState(prev => ({ count: prev.count + 1 }));
  };

  decrement = () => {
    this.setState(prev => ({ count: prev.count - 1 }));
  };

  sayWelcome = (msg) => {
    alert(`Welcome ${msg}`);
  };

  handleSynthetic = (e) => {
    alert('I was clicked');
  };

  render() {
    return (
      <div style={{ marginBottom: '30px' }}>
        <h2>Counter</h2>
        <p>Value: {this.state.count}</p>
        <button onClick={this.increment}>Increment (with
hello)</button>{' '}
        <button onClick={this.decrement}>Decrement</button>{'
'}
```

```
        <button onClick={() => this.sayWelcome('Guest')}>Say
Welcome</button>{' '}
        <button onClick={this.handleSynthetic}>Synthetic
OnPress</button>
      </div>
    );
  }
}

export default Counter;
```

src/components/CurrencyConvertor.js

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

function CurrencyConvertor() {
  const [rupees, setRupees] = useState('');
  const [euro, setEuro] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    // simplistic conversion: assume 1 Euro = 90 INR
    const converted = parseFloat(rupees) / 90;
    if (!isNaN(converted)) {
      setEuro(converted.toFixed(2));
    }
  };

  return (
    <div>
      <h2>Currency Convertor (INR to EUR)</h2>
      <form onSubmit={handleSubmit}>
        <label>
          INR:
          <input
            type="number"
            value={rupees}
            onChange={(e) => setRupees(e.target.value)}
          />
        </label>{' '}
        <button type="submit">Convert</button>
      </form>
      {euro && (
        <p>
          {rupees} INR = {euro} EUR
```

```
        </p>
      )}
    </div>
  );
}

export default CurrencyConvertor;
```

src/App.js

```
import React from 'react';
import Counter from './components/Counter';
import CurrencyConvertor from './components/
CurrencyConvertor';

function App() {
  return (
    <div style={{ padding: '20px' }}>
      <h1>Event Examples App</h1>
      <Counter />
      <CurrencyConvertor />
    </div>
  );
}

export default App;
```

Week 7

Lab 12: Conditional Rendering (`ticketbookingapp`)

Guest vs Logged-in user views toggle with Login/Logout.

src/components/GuestPage.js

```
import React from 'react';

function GuestPage({ onLogin }) {
  return (
    <div>
      <h2>Guest View</h2>
      <p>Browse flights, but you must login to book
tickets.</p>
      <button onClick={onLogin}>Login</button>
    </div>
  );
}

export default GuestPage;
```

src/components/UserPage.js

```
import React from 'react';

function UserPage({ onLogout }) {
  return (
    <div>
      <h2>User View</h2>
      <p>Book your tickets here.</p>
      <button onClick={onLogout}>Logout</button>
    </div>
  );
}

export default UserPage;
```

src/App.js

```
import React, { useState } from 'react';
import GuestPage from './components/GuestPage';
import UserPage from './components/UserPage';

function App() {
  const [loggedIn, setLoggedIn] = useState(false);
```

Week 7

```
  return (
    <div style={{ padding: '20px' }}>
      <h1>Ticket Booking App</h1>
      {loggedIn ? (
        <UserPage onLogout={() => setLoggedIn(false)} />
      ) : (
        <GuestPage onLogin={() => setLoggedIn(true)} />
      )}
    </div>
  );
}

export default App;
```

Week 7

Lab 13: Conditional Rendering Variants (`bloggerapp`)

Three detail components with multiple conditional patterns.

src/components/BookDetails.js

```
import React from 'react';

function BookDetails({ show }) {
  if (!show) return null;
  return (
    <div>
      <h3>Book Details</h3>
      <p>Title: React Patterns</p>
      <p>Author: Michael Chan</p>
    </div>
  );
}

export default BookDetails;
```

src/components/BlogDetails.js

```
import React from 'react';

function BlogDetails({ show }) {
  return show ? (
    <div>
      <h3>Blog Details</h3>
      <p>Topic: Conditional Rendering</p>
      <p>Posted on: August 1, 2025</p>
    </div>
  ) : null;
}

export default BlogDetails;
```

src/components/CourseDetails.js

```
import React from 'react';

function CourseDetails({ show }) {
  return (
    <div>
      {show && (
```

```
        <>
          <h3>Course Details</h3>
          <p>Course: Advanced React</p>
          <p>Duration: 8 weeks</p>
        </>
      )}
    </div>
  );
}

export default CourseDetails;
```

src/App.js

```
import React, { useState } from 'react';
import BookDetails from './components/BookDetails';
import BlogDetails from './components/BlogDetails';
import CourseDetails from './components/CourseDetails';

function App() {
  const [showBook, setShowBook] = useState(true);
  const [showBlog, setShowBlog] = useState(true);
  const [showCourse, setShowCourse] = useState(true);

  return (
    <div style={{ padding: '20px' }}>
      <h1>Blogger App</h1>

      <div style={{ marginBottom: '15px' }}>
        <button onClick={() => setShowBook(prev => !prev)}>
          Toggle Book Details
        </button>{' '}
        <button onClick={() => setShowBlog(prev => !prev)}>
          Toggle Blog Details
        </button>{' '}
        <button onClick={() => setShowCourse(prev => !prev)}>
          Toggle Course Details
        </button>
      </div>

      {/* Multiple conditional rendering techniques */}
      <BookDetails show={showBook} />
      <BlogDetails show={showBlog} />
      <CourseDetails show={showCourse} />
    </div>
```

```
  );
}

export default App;
```

To run each lab:

```
npm install
npm start
```

Open `http://localhost:3000` for the respective app.