3 ii)

<!doctype html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Simple Blog Reader</title>

    <style>

      body{font-family:system-ui, -apple-system, "Segoe UI", Roboto, Arial;max-width:900px;margin:24px auto;padding:0 16px;color:#222}

      h1{margin:0 0 12px}

      .post{border-bottom:1px solid #eee;padding:12px 0}

      .post h3{margin:0}

      .meta{color:#666;font-size:.9rem;margin-bottom:8px}

      button{margin-top:8px;padding:6px 10px}

      .detail{background:#fafafa;padding:12px;border:1px solid #eee;margin-top:12px}

    </style>

  </head>

  <body>

    <h1>Tiny Blog Reader</h1>

    <div id="root"></div>

    <!-- React -->

    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

    <script>

      const { useState } = React;

      // sample posts embedded so no server needed

      const POSTS = [

        { id: '1', title: 'Welcome to Tiny Blog', author: 'Mohan', date: '2025-10-13', body: 'This is a tiny blog built for teaching React. Click a post to read the full content.The command blog.html was not found, but does exist in the current location. Windows PowerShell does not load commands from the current location by default. If you trust this command, instead typePS E:\MERN\react\rblog> cd blog.html' },

        { id: '2', title: 'React Basics', author: 'Mohan', date: '2025-10-12', body: 'React uses components and state. Here we keep things very small and simple.' },

        { id: '3', title: 'Fetch Later', author: 'Mohan', date: '2025-10-11', body: 'Later you can replace the embedded posts with fetch() to an API or static JSON file.' }

      ];

      function PostList({ onOpen }) {

        return React.createElement('div', null,

          POSTS.map(p =>

            React.createElement('div', { key: p.id, className: 'post' },

              React.createElement('h3', null, p.title),

              React.createElement('div', { className: 'meta' }, `${p.author} • ${p.date}`),

              React.createElement('p', null, p.body.length > 120 ? p.body.slice(0,120) + '...' : p.body),

              React.createElement('button', { onClick: () => onOpen(p) }, 'Read')

            )

          )

        );

      }

      function PostDetail({ post, onBack }) {

        if (!post) return null;

        return React.createElement('div', { className: 'detail' },

          React.createElement('h2', null, post.title),

          React.createElement('div', { className: 'meta' }, `${post.author} • ${post.date}`),

          React.createElement('div', { style: { marginTop: 12 } }, post.body),

          React.createElement('div', null, React.createElement('button', { onClick: onBack }, '← Back to list'))

        );

      }

      function App() {

        const [open, setOpen] = useState(null);

        return React.createElement('div', null,

          React.createElement('div', null,

            React.createElement('h2', null, 'Posts'),

            React.createElement(PostList, { onOpen: setOpen })

          ),

          React.createElement(PostDetail, { post: open, onBack: () => setOpen(null) })

        );

      }

      ReactDOM.createRoot(document.getElementById('root')).render(React.createElement(App));

    </script>

  </body>

</html>

Command - .\filename.html

3iii) React Router

<!doctype html>

<html>

<head>

  <meta charset="utf-8" />

  <title>Simple React Hash Router (No react-router)</title>

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <style>

    body{font-family:Arial,Helvetica,sans-serif;margin:20px;background:#f7f7f7}

    .container{max-width:820px;margin:0 auto;background:#fff;padding:20px;border-radius:8px;box-shadow:0 2px 6px rgba(0,0,0,0.06)}

    header h1{margin:0 0 8px 0;font-size:22px}

    nav{margin-top:10px}

    nav a{margin-right:12px;text-decoration:none;color:#0077cc;padding:6px 8px;border-radius:4px}

    nav a.active{background:#0077cc;color:#fff}

    .page{margin-top:18px;padding:14px;border:1px solid #e6e6e6;border-radius:6px;background:#fff}

    footer{margin-top:18px;font-size:12px;color:#666}

  </style>

</head>

<body>

  <div class="container">

    <header>

      <h1>SIMPLE REACT HASH ROUTER</h1>

      <div>Works without react-router. Use links to navigate (hash-based).</div>

    </header>

    <div id="root"></div>

    <footer>

      Tip: Links use the hash (e.g. <code>#/about</code>) so this works from the file system or a server.

    </footer>

  </div>

  <!-- React UMD -->

  <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

  <script>

    (function () {

      const e = React.createElement;

      const { useState, useEffect } = React;

      // Helper: read current route from location.hash

      function getRoute() {

        const h = window.location.hash || '#/';

        // remove leading '#' and ensure leading '/'

        const path = h.slice(1);

        return path === '' ? '/' : path;

      }

      function Nav(props) {

        const route = props.route;

        function link(name, path) {

          const isActive = route === path;

          const style = { textDecoration: 'none', color: isActive ? '#fff' : '#0077cc', background: isActive ? '#0077cc' : 'transparent', padding: '6px 8px', borderRadius: '4px' };

          return e('a', { href: '#' + path, key: path, style }, name);

        }

        return e('nav', null, link('Home', '/'), link('About', '/about'), link('Contact', '/contact'));

      }

      function Home() {

        return e('div', { className: 'page' },

          e('h2', null, 'Home Page'),

          e('p', null, 'Welcome. This is the Home page in the simple hash-router demo.'),

          e('p', null, 'You can open this file directly in Chrome; the links use #/ routes.')

        );

      }

      function About() {

        return e('div', { className: 'page' },

          e('h2', null, 'About Page'),

          e('p', null, 'This example demonstrates routing implemented with hashchange and React.'),

          e('p', null, 'No bundler or extra installs required.')

        );

      }

      function Contact() {

        return e('div', { className: 'page' },

          e('h2', null, 'Contact Page'),

          e('p', null, 'Email: student@example.com'),

          e('p', null, 'Phone: +91-XXXXXXXXXX')

        );

      }

      function NotFound() {

        return e('div', { className: 'page' },

          e('h2', null, '404 — Page not found'),

          e('p', null, 'The requested page was not found. Use the navigation above.')

        );

      }

      function App() {

        const [route, setRoute] = useState(getRoute());

        useEffect(() => {

          function onHashChange() {

            setRoute(getRoute());

          }

          window.addEventListener('hashchange', onHashChange);

          // ensure route is synced if file opened w/o hash

          if (!window.location.hash) {

            window.location.hash = '#/';

          } else {

            setRoute(getRoute());

          }

          return () => window.removeEventListener('hashchange', onHashChange);

        }, []);

        let pageElem;

        if (route === '/') pageElem = e(Home);

        else if (route === '/about') pageElem = e(About);

        else if (route === '/contact') pageElem = e(Contact);

        else pageElem = e(NotFound);

        return e('div', null, e(Nav, { route }), pageElem);

      }

      const root = ReactDOM.createRoot(document.getElementById('root'));

      root.render(e(App));

    })();

  </script>

</body>

</html>

4 i) Bookstore API

const express = require('express');

const app = express();

app.use(express.json());

// temporary data (acts as our “database”)

let books = [

  { id: 1, title: "Hindustan dairies", author: "cyber students" }

];

// show all books

app.get('/books', (req, res) => {

  res.json(books);

});

// add a new book

app.post('/books', (req, res) => {

  const newBook = {

    id: books.length + 1,

    title: req.body.title,

    author: req.body.author

  };

  books.push(newBook);

  res.status(201).json(newBook);

});

// start the server

app.listen(3000, () => {

  console.log('Bookstore API running at http://localhost:3000');

});

Command – npm init -y

npm install express

node server.js

4 ii) mongodb

// server.js — Smallest MongoDB connection example

const mongoose = require('mongoose');

// 1 Connect to local MongoDB (make sure mongod is running)

mongoose.connect('mongodb://127.0.0.1:27017/testdb2')

  .then(() => console.log('Connected to MongoDB successfully!'))

  .catch(err => console.error(' Connection error:', err));

// 2 Define a simple schema (like a table)

const userSchema = new mongoose.Schema({

  name: String,

  age: Number

});

// 3 Create a model

const User = mongoose.model('User', userSchema);

// 4 Insert one sample record

async function runDemo() {

  const user = new User({ name: 'Cyber2', age: 25 });

  await user.save();

  console.log(' User saved to MongoDB!');

  mongoose.connection.close();

}

runDemo();

commands : npm init -y

npm install express

npm install mongoose

npm server.js

4 iii) jwt authentication

// server.js — Simplest JWT Authentication Example

const express = require('express');

const jwt = require('jsonwebtoken');

const app = express();

app.use(express.json());

const SECRET\_KEY = 'my\_secret\_key'; // only for demo

const USER = { username: 'admin', password: '1234' }; // hardcoded user

// 🟢 Login route

app.post('/login', (req, res) => {

  const { username, password } = req.body;

  if (username === USER.username && password === USER.password) {

    const token = jwt.sign({ username }, SECRET\_KEY, { expiresIn: '1h' });

    return res.json({ token });

  }

  res.status(401).json({ message: 'Invalid credentials' });

});

// 🟡 Protected route

app.get('/protected', (req, res) => {

  const authHeader = req.headers.authorization;

  if (!authHeader) return res.status(401).json({ message: 'Missing token' });

  const token = authHeader.split(' ')[1];

  try {

    const decoded = jwt.verify(token, SECRET\_KEY);

    res.json({ message: `Welcome ${decoded.username}, you accessed a protected route!` });

  } catch {

    res.status(403).json({ message: 'Invalid or expired token' });

  }

});

app.listen(3000, () => console.log(' JWT demo running at http://localhost:3000'));

command

npm init -y

npm install express

npm install jsonwebtoken

node server.js

5 i) front and back end integration

server.js

// server.js — serve index.html in the same folder

const express = require('express');

const path = require('path');

const app = express();

const PORT = 3000;

// API example

app.get('/api/message', (req, res) => res.json({ message: 'Hello from Express API!' }));

// serve index.html for the root and any unmatched routes

app.get('/', (req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.use((req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.listen(PORT, () => console.log(`Server running: http://localhost:${PORT}`));

index.html

<!doctype html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Part 1 — React + Express</title>

  </head>

  <body>

    <div id="root"></div>

    <!-- React from CDN -->

    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

    <script>

      const e = React.createElement;

      function App() {

        const [msg, setMsg] = React.useState('Loading...');

        React.useEffect(() => {

          fetch('/api/message')

            .then(r => r.json())

            .then(j => setMsg(j.message))

            .catch(() => setMsg('Error fetching message'));

        }, []);

        return e('div', { style: { fontFamily: 'sans-serif', padding: 20 } },

          e('h2', null, 'Part 1 — React + Express Integration'),

          e('p', null, 'API says:'),

          e('pre', null, msg)

        );

      }

      ReactDOM.createRoot(document.getElementById('root')).render(e(App));

    </script>

  </body>

</html>

Commands :

npm init -y

npm install express

npm install react

node server.js

5 ii) full stack task manager

Index.html

<!doctype html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Part 1 — React + Express</title>

  </head>

  <body>

    <div id="root"></div>

    <!-- React from CDN -->

    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

    <script>

      const e = React.createElement;

      function App() {

        const [msg, setMsg] = React.useState('Loading...');

        React.useEffect(() => {

          fetch('/api/message')

            .then(r => r.json())

            .then(j => setMsg(j.message))

            .catch(() => setMsg('Error fetching message'));

        }, []);

        return e('div', { style: { fontFamily: 'sans-serif', padding: 20 } },

          e('h2', null, 'Part 1 — React + Express Integration'),

          e('p', null, 'API says:'),

          e('pre', null, msg)

        );

      }

      ReactDOM.createRoot(document.getElementById('root')).render(e(App));

    </script>

  </body>

</html>

Server.js

// server.js — serve index.html in the same folder

const express = require('express');

const path = require('path');

const app = express();

const PORT = 3000;

// API example

app.get('/api/message', (req, res) => res.json({ message: 'Hello from Express API!' }));

// serve index.html for the root and any unmatched routes

app.get('/', (req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.use((req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.listen(PORT, () => console.log(`Server running: http://localhost:${PORT}`));

commands

npm init -y

npm install express

npm install react

node server.js

5 iii) simple deployment

Server.js

const express = require('express');

const path = require('path');

const app = express();

const PORT = process.env.PORT || 3000;

app.get('/', (req, res) => res.sendFile(path.join(\_\_dirname, 'index.html')));

app.get('/api/message', (req, res) => res.json({ message: 'Deployed successfully! 🚀' }));

app.listen(PORT, () => console.log(` App running on port ${PORT}`));

index.html

<!doctype html><html><head><meta charset="utf-8"/><title>Deployment Demo</title></head>

<body>

  <h1>🚀 Node.js Deployment Demo</h1>

  <button id="btn">Check API Message</button>

  <pre id="output"></pre>

  <script>

    document.getElementById("btn").onclick = async () => {

      const res = await fetch("/api/message");

      const data = await res.json();

      document.getElementById("output").textContent = JSON.stringify(data, null, 2);

    };

  </script>

</body></html>

Commands (local)

npm init -y

npm install express

npm install react

node server.js

commands (cloud)

github commands

git init -y

git add .

git commit -m “Initial commit – depdemo (name)”

git remote add origin <https://github.com/NivasLeo/depdem.git> (url)

git branch -M main

git push -u origin main

Render

Go to render.com

Click new 🡪 webservice

Build command 🡪 npm install

Start command 🡪 node server.js

Click deploy web service button

6. task manager in cloud

Index.html

<!doctype html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Part 1 — React + Express</title>

  </head>

  <body>

    <div id="root"></div>

    <!-- React from CDN -->

    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>

    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

    <script>

      const e = React.createElement;

      function App() {

        const [msg, setMsg] = React.useState('Loading...');

        React.useEffect(() => {

          fetch('/api/message')

            .then(r => r.json())

            .then(j => setMsg(j.message))

            .catch(() => setMsg('Error fetching message'));

        }, []);

        return e('div', { style: { fontFamily: 'sans-serif', padding: 20 } },

          e('h2', null, 'Part 1 — React + Express Integration'),

          e('p', null, 'API says:'),

          e('pre', null, msg)

        );

      }

      ReactDOM.createRoot(document.getElementById('root')).render(e(App));

    </script>

  </body>

</html>

Server.js

// server.js — serve index.html in the same folder

const express = require('express');

const path = require('path');

const app = express();

const PORT = 3000;

// API example

app.get('/api/message', (req, res) => res.json({ message: 'Hello from Express API!' }));

// serve index.html for the root and any unmatched routes

app.get('/', (req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.use((req, res) => {

  res.sendFile(path.join(\_\_dirname, 'index.html'));

});

app.listen(PORT, () => console.log(`Server running: http://localhost:${PORT}`));

commands

npm init -y

npm install express

npm install react

node server.js

commands (cloud)

github commands

git init -y

git add .

git commit -m “Initial commit – depdemo (name)”

git remote add origin <https://github.com/NivasLeo/depdem.git> (url)

git branch -M main

git push -u origin main

Render

Go to render.com

Click new 🡪 webservice

Build command 🡪 npm install

Start command 🡪 node server.js

Click deploy web service button