

A REPORT
ON
EDUQUEST Learning Platform
Offered by the
Department Computer Science and Engineering
School of Engineering and Sciences
Submitted by
B.Kaivalya AP23110011311
S.Meghana AP23110011415
K.Pujitha AP23110011433
M.Jahnavi AP23110011445



SRM University-AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh – 522 240
December 2025

EDUQUEST(React)

Introduction:

Step into the future of digital learning — where education evolves beyond boundaries and transforms into an immersive, intelligent, and student-centric journey. EDUQUEST, our next-generation e-learning platform, redefines how learners interact with knowledge by blending modern web technologies with a seamless learning experience.

Designed with precision and powered by the dynamic capabilities of React.js, EDUQUEST stands as a revolutionary milestone in online education. Every component — from course browsing to assessments — has been crafted to offer clarity, speed, and engagement. Whether a student is exploring new subjects, attempting quizzes, or tracking their performance, EDUQUEST ensures a smooth, intuitive, and powerful interaction at every step.

At the heart of EDUQUEST lies a robust combination of React, modular UI design, and API-driven communication. This not only enhances the learning environment but also empowers administrators to manage courses, content, and student data with unmatched efficiency. Learners can dive into interactive modules, watch lessons, read notes, attempt assignments, and receive instant feedback — all within a beautifully designed interface that works flawlessly across desktops, tablets, and smartphones.

Bid farewell to rigid, outdated learning systems and embrace a platform where education adapts to the learner. With EDUQUEST, we bring innovation, accessibility, and personalization together, creating a space where every learner's journey feels unique, dynamic, and inspiring.

Scenario-Based Introduction :-

Imagine a student starting their day, anxious about an upcoming exam and unsure where to begin revising. Instead of juggling textbooks, printed notes, and dozens of bookmarks, they open EDUQUEST, their all-in-one digital learning companion.

The moment the platform loads, the student is welcomed with a clean and organized dashboard showing their enrolled courses, recently viewed lessons, and upcoming quizzes. With a single click, they return to the chapter they were studying the previous night. The content is neatly structured, easy to navigate, and supported by notes that help them quickly recall important concepts.

As they continue learning, the student decides to test their understanding. EDUQUEST instantly generates a quiz based on that topic. After submitting the answers, the platform provides immediate results along with explanations, helping the student clearly see where they stand and what needs more attention.

Later in the day, the student receives a notification—new study material has been uploaded by the instructor. Without any hassle, the student accesses it directly from the platform, eliminating the need to search across different apps or folders.

Whether used at home, during travel, or in a classroom, EDUQUEST ensures every student has quick access to lessons, notes, quizzes, and updates—all in one place. It transforms the daily learning routine into a smooth, organized, and deeply engaging experience that supports students at every step of their academic journey.

Target Audience:-

1. Students

Learners from schools, colleges, and universities who want easy access to study materials, notes, quizzes, and performance tracking—all in one organized platform.

2. Teachers & Instructors

Educators who need a simple and effective way to upload content, share notes, manage student progress, and create assessments.

3. Self-Learners

Individuals preparing for competitive exams or learning at their own pace, seeking structured modules, quick revision tools, and interactive learning resources.

4. Institutions

Educational centers looking for a centralized, digital solution to streamline teaching, content distribution, and student engagement.

Project Goals and Objectives

The primary goal of EDUQUEST is to create a smart, accessible, and interactive e-learning environment that enhances the teaching and learning experience. Our key objectives include:

1. User-Friendly Learning Interface

Design an intuitive and clean interface that allows students and teachers to navigate courses, notes, quizzes, and updates effortlessly, improving overall engagement.

2. Centralized Course Management

Provide a structured system for uploading, organizing, and managing educational content, ensuring students always find the right material at the right time.

3. Interactive Learning Experience

Offer quizzes, evaluations, instant feedback, and progress tracking to help students actively monitor and improve their understanding of each subject.

4. Efficient Teacher–Student Interaction

Enable instructors to easily share materials, upload updates, and communicate essential information, ensuring a smooth and connected learning workflow.

5. Accessibility Across Devices

Ensure EDUQUEST works seamlessly on desktops, tablets, and mobile devices, allowing users to learn anytime, anywhere.

Key Features

1. Structured Learning Modules

EDUQUEST provides neatly organized courses with chapters, notes, and study materials, helping students access everything they need in one place.

2. Digital Notes and Resources

Students can easily view and revise uploaded notes, while teachers can add new materials through a simple and efficient upload system.

3. Quizzes with Instant Feedback

The platform includes quick assessments that give immediate results and explanations, helping students track their understanding and improve effectively.

4. Student Progress Tracking

EDUQUEST monitors completed topics, quiz scores, and study activity, giving learners a clear picture of their academic progress.

5. Teacher/Admin Management Panel

Instructors can manage courses, upload resources, and organize content through a user-friendly dashboard, making teaching tasks more efficient.

6. Responsive and Accessible Design

The platform works smoothly across mobile, tablet, and desktop, allowing students to learn anytime, anywhere with ease.

PRE-REQUISITES

Developing the EDUQUEST e-learning platform using React.js requires a few essential tools and foundational skills:

1. Node.js and npm

Node.js provides the environment needed to run JavaScript outside the browser, while npm manages project dependencies.

You must install both before creating or running the EDUQUEST frontend.

Download: <https://nodejs.org/en/download/>

2. React.js Setup

React.js is the core library used to build EDUQUEST's user interface.

To create the project, follow these steps:

```
npm create vite@latest
```

```
cd project-name
```

```
npm install
```

```
npm run dev
```

This starts the development server and opens the app at <http://localhost:5173>.

3. Basic Web Development Skills

A working understanding of HTML, CSS, and JavaScript is important for building components, styling pages, and adding interactivity to the platform.

4. Version Control (Git)

Git helps track changes, collaborate with team members, and maintain clean code history.

Download Git: <https://git-scm.com/downloads>

Platforms like GitHub or GitLab can be used to host the repository.

5. Code Editor / IDE

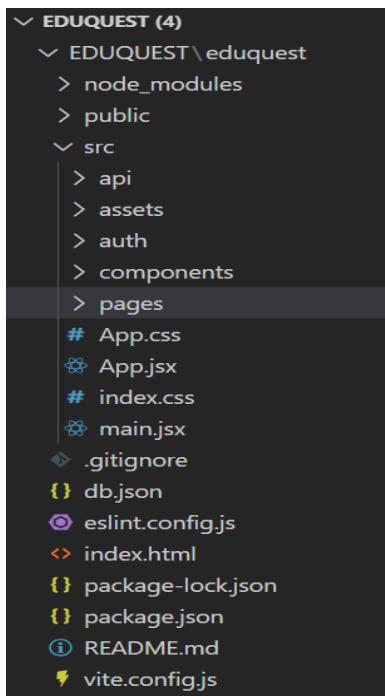
A powerful editor enhances productivity while developing EDUQUEST. Popular choices include:

Visual Studio Code – <https://code.visualstudio.com/download>

Sublime Text – <https://www.sublimetext.com/download>

WebStorm – <https://www.jetbrains.com/webstorm/download>

Project Structure :



The project structure can vary based on the framework or development approach used, but it is important to keep files organized in a clear and consistent manner to ensure easy maintenance and smooth collaboration among developers.

In EDUQUEST, the main structure includes the root application files that manage layout and routing:

App.jsx, App.css (src/App.jsx, src/App.css) :

These files belong to the main App component, which acts as the root of the React application. The App component controls the overall layout of EDUQUEST and contains the routing setup used to load different pages—such as Courses, Notes, Quizzes, and the Admin Panel—based on the selected route.

Before understanding how EDUQUEST works internally, it is helpful to view the platform in action. The project follows a simple and structured flow that begins with user interaction and ends with smooth navigation through courses, notes, quizzes, and admin features.

Project Demo

Experience how EDUQUEST functions through this live demonstration:

Project Demonstration Video:

<https://drive.google.com/file/d/1ZSErS0DnJEbpzMgI69BLO40TR08KsMWz/view?usp=sharing>

Project Code Explanation Video:

<https://drive.google.com/file/d/1HkTc8ekMFJFDhdd94i9rCMTue1GPxBg/view?usp=drivesdk>

Project Code Files:

<https://github.com/pujitha244/EduQuest-React-App>

Milestone 1: Project Setup and Configuration

1. Install Required Tools and Software

Begin by opening your project folder and installing the essential tools needed for building the EDUQUEST platform. This project uses:

React.js
Material-UI (MUI)
React Icons
React-codemirror2
Node.js & npm (for running and installing project)

These libraries help create a smooth, interactive, and visually consistent learning platform.

Reference Links

- React Installation Guide: <https://react.dev/learn/installation>
- UI Libraries Reference: <https://react-bootstrap-v4.netlify.app/getting-started/introduction/>

Milestone 2: Web Development

1. Setup React Application

- Created the EDUQUEST frontend project using React and Vite for fast development and modular coding.
- Installed required libraries such as Material UI, React Icons, and React-Codemirror2 to support UI components and interactive features.
- Installed JSON Server to store and manage Courses, Notes, and Quiz data used in the platform.
- Set up the project workspace in Visual Studio Code and initialized the core folder structure for development.

App.jsx Component – Code Description

- App.jsx functions as the root component of the EDUQUEST application and contains the main page layout structure.

- Common layout components like Navbar and Footer are imported here to ensure they appear across all pages.
- The App component acts as a wrapper to display all core modules such as Home, Courses, Notes, Quizzes, Login, Register, and Admin Panel inside a unified UI.
- Because React renders App.jsx in the browser, EDUQUEST works as a Single Page Application (SPA), ensuring smooth navigation without full page reloads.
- App.jsx is exported as the default component, allowing it to launch the entire application through main.jsx.

2. Design UI Components

- Create reusable components such as Navbar, Course Cards, Notes Viewer, and Quiz Elements.
- Implement layout and styling using CSS and UI libraries to ensure a clean and user-friendly interface.
- Add navigation to allow smooth movement between pages like Home, Courses, Notes, Quizzes, and Admin Panel

3. Implement Frontend Logic

i. API Integration

```
EDUQUEST > eduquest > src > api > JS api.js > ...  
1  import axios from "axios";  
2  
3  const api = axios.create({  
4    |   baseURL: "http://localhost:5000",  
5    |   timeout: 8000,  
6  | });  
7  
8  export default api;  
9
```

Code Description :

- Imports axios (or fetch) to perform HTTP requests.
- Creates reusable API functions for retrieving Courses, Notes, and Quiz data from db.json.
- Sends GET requests to backend endpoints like /courses, /notes, /quiz.
- Returns fetched data to the calling UI components for display.
- Simplifies data communication by separating backend logic from UI code.
- Updates UI whenever database content changes — improving dynamic behavior.
- Prevents duplicated code by managing all API calls in one place.
- Makes application scalable — new features can reuse same pattern.
- Handles API errors for secure and smooth data transactions.
- Exports API functions so they can be imported into pages like Courses.js

ii.Data Binding using React Hooks :

```
import { useState } from "react";
import { useAuth } from "../auth/AuthContext";
import { Link, useNavigate } from "react-router-dom";

export default function CreateAccount() {
  const { register } = useAuth(); // <-- register MUST come from context
  const navigate = useNavigate();

  const [form, setForm] = useState({
    name: "",
    email: "",
    password: "",
  });

  const [error, setError] = useState("");

  const handleChange = (e) => {
    setForm({ ...form, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setError("");

    try {
      await register(form); // <-- register() called
      navigate("/"); // redirect home
    } catch (err) {
      setError(err.message || "Registration failed.");
    }
  };

  return (
    <div className="max-w-md mx-auto mt-10 bg-white p-6 rounded-xl shadow">
      <h2 className="text-xl font-bold text-slate-800 mb-4">Create Account</h2>

      {error && (
        <p className="mb-3 text-red-600 text-sm bg-red-50 border border-red-200 p-2 rounded">
          {error}
        </p>
      )}
    </div>
  );
}
```

```

export default function CreateAccount() {
  <form onSubmit={handleSubmit} className="space-y-4">
    <div>
      <label className="text-sm text-slate-600">Full Name</label>
      <input
        type="text"
        name="name"
        value={form.name}
        onChange={handleChange}
        className="w-full px-3 py-2 border rounded-lg"
      />
    </div>
    <div>
      <label className="text-sm text-slate-600">Email</label>
      <input
        type="email"
        name="email"
        value={form.email}
        onChange={handleChange}
        className="w-full px-3 py-2 border rounded-lg"
      />
    </div>
    <div>
      <label className="text-sm text-slate-600">Password</label>
      <input
        type="password"
        name="password"
        value={form.password}
        onChange={handleChange}
        className="w-full px-3 py-2 border rounded-lg"
      />
    </div>
    <button className="w-full bg-indigo-600 text-white py-2 rounded-lg font-semibold hover:bg-indigo-700">
      Create Account
    </button>
  </form>

  <p className="text-center text-sm mt-4">
    Already have an account?{" "}
    <Link to="/login" className="text-indigo-600 underline">
      Sign in
    </Link>
  </p>
</div>
);
}

```

Code description : (CreateAccount.jsx)

- Imports useState from React to store form field values like username, email, and password.
- Imports CSS and UI elements for building a visually clean registration form.
- Defines the CreateAccount functional component that handles sign-up operations.
- Uses controlled input components to dynamically update state when user types.
- Includes a form submit event handler to prevent default reload behavior.

- Sends a POST request to the backend server (JSON server) to store new user account data.
- Performs input validation to ensure the fields are properly filled before submission.
- Displays alerts/messages to indicate registration success or failure.
- Redirects users to Login or Home page after successful account creation.
- Supports accessibility and responsiveness for different screen sizes.
- Exports CreateAccount component so it can be used inside App.jsx routing.

iii. Routing & Navigation :

```
import { Routes, Route } from "react-router-dom";
// Public pages
import Navbar from "../components/Navbar";
import Footer from "../components/Footer";
import Register from "../pages/Register";
import Home from "../pages/Home";
import About from "../pages/About";
import Courses from "../pages/Courses";
import CourseDetails from "../pages/CourseDetails";
import Team from "../pages/Team";
import Testimonials from "../pages/Testimonials";
import PricingCheckout from "../pages/PricingCheckout";
import Blog from "../pages/Blog";
import BlogDetails from "../pages/BlogDetails";
import Contact from "../pages/Contact";
import Enrolled from "../pages/Enrolled";
import NotFound from "../pages/NotFound";
import Login from "../pages/Login";
import LearnCourse from "../pages/LearnCourse";
import Certificate from "../pages/Certificate";
// Admin pages
import AdminCourses from "../pages/admin/AdminCourses";
import AdminBlog from "../pages/admin/AdminBlog";
import AdminTeam from "../pages/admin/AdminTeam";
import AdminTestimonials from "../pages/admin/AdminTestimonials";
import AdminLessons from "../pages/admin/AdminLessons";
// Auth
import ProtectedRoute from "../auth/ProtectedRoute";
export default function App() {
  return (
    <div className="min-h-screen flex flex-col bg-purple-50">
      <Navbar />
      <main className="flex-1">
        <div className="max-w-6xl mx-auto px-4 py-8">
          <Routes>
            { /* PUBLIC ROUTES */ }
            <Route path="/" element={ <Home /> } />
            <Route path="/about" element={ <About /> } />
            <Route path="/register" element={ <Register /> } />
            { /* Courses */ }
            <Route path="/courses" element={ <Courses /> } />
            <Route path="/courses/:id" element={ <CourseDetails /> } />
            { /* Learning Page */ }
            <Route path="/learn/:courseId" element={ <LearnCourse /> } />
            { /* Certificates */ }
            <Route path="/certificate/:courseId" element={ <Certificate /> } />
            { /* Other Public Pages */ }
            <Route path="/team" element={ <Team /> } />
          </Routes>
        </div>
      </main>
      <Footer />
    </div>
  );
}
```

```

<Route path="/team" element={<Team />} />
<Route path="/testimonials" element={<Testimonials />} />
<Route path="/pricing/:id/checkout" element={<PricingCheckout />} />
<Route path="/blog" element={<Blog />} />
<Route path="/blog/:id" element={<BlogDetails />} />
<Route path="/enrolled" element={<Enrolled />} />
<Route path="/contact" element={<Contact />} />
<Route path="/login" element={<Login />} />
{/* ===== ADMIN ROUTES ===== */}
<Route
  path="/admin/courses"
  element={
    <ProtectedRoute roles={["admin"]}>
      <AdminCourses />
    </ProtectedRoute>
  }
/>
<Route
  path="/admin/blog"
  element={
    <ProtectedRoute roles={["admin"]}>
      <AdminBlog />
    </ProtectedRoute>
  }
/>
<Route
  path="/admin/team"
  element={
    <ProtectedRoute roles={["admin"]}>
      <AdminTeam />
    </ProtectedRoute>
  }
/>
<Route
  path="/admin/testimonials"
  element={
    <ProtectedRoute roles={["admin"]}>
      <AdminTestimonials />
    </ProtectedRoute>
  }/>
<Route
  path="/admin/lessons"
  element={
    <ProtectedRoute roles={["admin"]}>
      <AdminLessons />
    </ProtectedRoute>
  }/>

```

```

      <ProtectedRoute roles={["admin"]}>
        <AdminLessons />
      </ProtectedRoute> }/> { /* 404 Page */ }
    <Route path="*"
      element={<NotFound />} /> </Routes>
  </div> </main><Footer /> </div>
);
}

```

Code Description – App.jsx

- Imports BrowserRouter, Routes, and Route from React Router to handle navigation.
- Imports major page components such as Home, Courses, Notes, Quiz, Login, Create Account, and Admin.
- Defines the App functional component as the root of the application.
- Wraps all content inside <BrowserRouter> to enable SPA (Single Page Application) navigation.
- Uses <Routes> and multiple <Route> elements to map URLs to corresponding page components.
- Ensures that each navigation click loads the correct module without reloading the entire page.
- Contains a Navbar import (if used) to keep menu accessible across all pages.
- Maintains a clean and modular structure for future page additions.
- Ensures the entire EDUQUEST application renders inside this root wrapper.
- Exports the App component as default so it can be imported into main.jsx.

iv. UI Components Implementation :

```
import { Link, NavLink } from "react-router-dom";
import { useAuth } from "../auth/AuthContext";
import { useState, useRef, useEffect } from "react";

const base =
  "text-sm font-medium hover:text-purple-600 transition-colors";
const active = "text-purple-600 font-semibold";

export default function Navbar() {
  const { user, role, logout } = useAuth();

  return (
    <header className="border-b border-purple-100 glass sticky top-0 z-50">
      <nav className="max-w-6xl mx-auto px-4 py-3 flex items-center justify-between gap-4">
        {/* Logo */}
        <Link to="/" className="flex items-center gap-2">
          <div className="h-8 w-8 rounded-xl bg-purple-600 text-white flex items-center justify-center text-xs font-bold">
            EQ
          </div>
          <div>
            <p className="text-sm font-bold text-slate-900">EduQuest</p>
            <p className="text-[11px] text-slate-500">Explore • Learn • Thrive</p>
          </div>
        </Link>

        {/* Main Navigation */}
        <div className="hidden md:flex items-center gap-4">
          <NavItem to="/">Home</NavItem>
          <NavItem to="/about">About</NavItem>
          <NavItem to="/courses">Courses</NavItem>

          <NavItem to="/blog">Blog</NavItem>
          <NavItem to="/team">Team</NavItem>
          <NavItem to="/testimonials">Testimonials</NavItem>
          <NavItem to="/enrolled">Enrolled</NavItem>
          <NavItem to="/contact">Contact</NavItem>
        </div>

        {/* RIGHT SIDE */}
        <div className="flex items-center gap-3">
          {/* Admin dropdown only for admin */}
          {role === "admin" && <AdminMenu />}

          {/* Greeting */}
          {user && (
```

```

    )
    /* Authentication Buttons */
    {!user ? (
      <div className="flex items-center gap-2">
        <Link
          to="/login"
          className="text-xs px-3 py-1.5 rounded-xl border border-slate-300 text-slate-700 hover:bg-slate-100"
        >
          Login
        </Link>

        <Link
          to="/register"
          className="text-xs px-3 py-1.5 rounded-xl bg-purple-600 text-white hover:bg-purple-700 shine"
        >
          Sign Up
        </Link>
      </div>
    ) : (
      <button
        onClick={logout}
        className="text-xs px-3 py-1.5 rounded-xl bg-red-500 text-white hover:bg-red-600"
      >
        Logout
      </button>
    )
  </div>
</nav>
</header>
);
}

```

```

/*****
* NAV LINK
*****/
function NavItem({ to, children }) {
  return (
    <NavLink
      to={to}
      className={({ isActive }) => base + (isActive ? " " + active : "")}
    >
      {children}
    </NavLink>
  );
}

```

```

/* ADMIN MENU (CLICK DROPDOWN)
***** */
function AdminMenu() {
  const [open, setOpen] = useState(false);
  const menuRef = useRef(null);
  // Close dropdown when clicking outside
  useEffect(() => {
    const handleClickOutside = (e) => {
      if (menuRef.current && !menuRef.current.contains(e.target)) {
        setOpen(false);
      }
    };
    document.addEventListener("mousedown", handleClickOutside);
    return () => {
      document.removeEventListener("mousedown", handleClickOutside);
    };
  }, []);
  return (
    <div className="relative" ref={menuRef}>
      <button
        onClick={() => setOpen((prev) => !prev)}
        className="text-xs px-3 py-1.5 rounded-xl bg-purple-600 text-white hover:bg-purple-700 shine"
        >
        Admin Panel ▾
      </button>

      {/* Dropdown */}
      {open && (
        <div className="absolute right-0 mt-2 w-56 bg-white border border-slate-200 rounded-xl shadow-xl py-2 z-50">
          <DropdownLink to="/admin/courses">Manage Courses</DropdownLink>
          <DropdownLink to="/admin/blog">Manage Blog</DropdownLink>
          <DropdownLink to="/admin/team">Manage Team</DropdownLink>
          <DropdownLink to="/admin/testimonials">Manage Testimonials</DropdownLink>
          <DropdownLink to="/admin/lessons">Manage Lessons</DropdownLink>
        </div>
      )}
    </div>
  );
}

function DropdownLink({ to, children }) {
  return (
    <Link
      to={to}
      className="block px-4 py-2 text-sm text-slate-700 hover:bg-slate-100 rounded-lg"
      >
      {children}
    </Link>
  );
}

```

Code Description :

- Imports Link or NavLink from React Router to enable page navigation without reloading.
- Imports React and required UI elements (e.g., buttons, icons, CSS styles) for visual layout.
- Defines the Navbar functional component used across multiple pages.
- Displays navigation menu items such as Home, Courses, Notes, Quiz, Login, Signup, Admin (based on your app structure).
- Uses semantic HTML tags like <nav>, , for organized and accessible structure.
- Maintains a consistent header on every screen to improve user experience and ease of navigation.

- Uses CSS classes or styled-components to style menu items including hover effects and active highlighting.
- Implements a responsive layout, adapting navigation bar elements on different screen sizes like mobile/tablet.
- Ensures a smooth learning flow by letting students easily switch between modules.
- Keeps routing logic separate from design, improving modularity and maintainability.
- Can conditionally render options (e.g., show Login when logged out & Logout when logged in).
- Enhances usability with icon support where required (like menu icons or profile icons).
- Exports the Navbar component so it can be imported into App.jsx and displayed across all pages.

5. System Flow :

To run the EDUQUEST platform, both the React frontend and the JSON server must be started.

Start React App:

npm start

or (if using Vite)

npm run dev

Start JSON Server:

json-server --watch db.json --port 3000

This will launch the application in the browser and start the backend service for fetching data.

Outputs :

Sign in Page

EQ

EduQuest

Explore • Learn • Thrive

HomeAboutCoursesBlogTeamTestimonialsEnrolledContact

LoginSign Up

Sign in to EduQuest

Students can use any email and password. Admin has a specific login.

Email

you@example.com

Password

Sign in

Home Page

EQ

EduQuest

Explore • Learn • Thrive

HomeAboutCoursesBlogTeamTestimonialsEnrolledContact

Hi, abcLogout

LIMITLESS LEARNING

Unlock your potential
with EduQuest.

Join thousands of learners on a journey of discovery. Access top-tier courses, receive mentorship from industry experts, and build the practical skills you need to shape your future.

Browse CoursesHow EduQuest Works

Industry-Recognized

Curriculum designed by top experts.

Self-Paced Learning

Study on your schedule, anytime.

150+

Active Courses

10k+

Students Enrolled

4.9/5

Student Rating

24/7

Mentor Support

Start your journey today

Whether you want to advance your career or explore a new hobby, EduQuest provides the tools and community to help you succeed.

Why learners choose EduQuest

We focus on providing a seamless, interactive, and high-quality learning experience for everyone.

Expert-Led Instruction

Learn directly from professionals who have worked at top-tier companies and mastered their fields.

Interactive Projects

Don't just watch videos. Build real-world projects that reinforce your learning and build your portfolio.

Career Advancement

Earn certificates upon completion and gain the confidence to ask for a promotion or switch careers.

About Page

EQ

EduQuest

Explore • Learn • Thrive

HomeAboutCoursesBlogTeamTestimonialsEnrolledContact

Hi, abcLogout

About EduQuest

Empowering learners worldwide with cutting-edge courses, expert mentorship, and a community dedicated to lifelong growth.

Our story

EduQuest began with a simple yet powerful idea: that quality education should be accessible to everyone, regardless of their background or location. We recognized a growing gap between traditional academic learning and the practical skills demanded by the modern industry.

What started as a small initiative has grown into a comprehensive learning hub. We brought together passionate educators, industry experts, and technology enthusiasts to build a platform where curiosity meets opportunity.

Today, EduQuest stands as a bridge to your future. Whether you are looking to upskill for a promotion, pivot to a new career, or simply explore a new passion, we provide the structured path and resources to help you thrive.

Our mission

To democratize education by providing high-quality, affordable, and industry-relevant resources that bridge the gap between theoretical knowledge and real-world application.

Our vision

To become the world's leading destination for online learning, fostering a global community of innovators, thinkers, and leaders who are equipped to shape the future.

Courses Page

EQ

EduQuest


Explore • Learn • Thrive

HomeAboutCoursesBlogTeamTestimonialsEnrolledContact

Hi, abcLogout

EduQuest Courses

Explore expert-designed courses to accelerate your learning journey.




React

React Basics

Beginner • 4 weeks • 10 lessons

Learn React fundamentals including components, props, state, and hooks. Ideal for beginners entering frontend development.

\$0




Full-Stack Development

Advanced JavaScript

Intermediate • 6 weeks • 15 lessons

Master modern JavaScript features including closures, async/await, prototypes, modules, and performance optimization.

\$59



Full-Stack Fundamentals

Intermediate • 8 weeks • 20 lessons

Understand frontend, backend, APIs, and databases. Build end-to-end full-stack applications using modern tools.

\$79

© 2025 EduQuest. All rights reserved.

Built with React, Tailwind CSS & JSON Server.

Blog Page

EQ

EduQuest

Explore • Learn • Thrive

Home

About

Courses

Blog

Team

Testimonials

Enrolled

Contact

Hi, abc

Logout

EduQuest Blog

Explore learning tips, roadmap guides, and career insights.

How to Stay Consistent While Learning Online

by Alice Johnson

Consistency is the key to mastering any skill. Set fixed study hours, reduce distractions, and break your learning into weekly milestones. Track your progress and reward yourself ...

Top 5 Skills Web Developers Need in 2025

by Rahul Mehta

To stay competitive in 2025, focus on: 1. JavaScript mastery 2. React & component-based development 3. TypeScript 4. API integrations 5. Problem-solving & communication

Team Page

EQ

EduQuest

Explore • Learn • Thrive

Home

About

Courses

Blog

Team

Testimonials

Enrolled

Contact

Hi, abc

Logout

Meet the EduQuest team

Educators, mentors, and content designers focused on making learning more practical and enjoyable.

M

Meghana

Lead Instructor

Frontend engineer with 8+ years of experience teaching React, JavaScript, and UI architecture.

J

Jahnavi

Curriculum Designer

Designs structured learning paths and real-world project-based content for developers.

K

Kaivalya

Student Success Coach

Helps learners stay consistent and achieve their learning goals through structured mentoring.

P

pujitha

Mentor

100+ Projects on React based applications

Testimonials Page

What learners say

Feedback from students and professionals who used EduQuest-style platforms to refine their skills.

"EduQuest helped me build a strong foundation in web development with simple explanations and good examples."

— Sara

"The platform is beginner-friendly and the content is very easy to follow. Highly recommended!"

— Imran

"The practical examples and mini-projects helped me understand concepts deeply."

— Meera

Enrolled Page

My Enrolled Courses

Continue where you left off and track what you've completed.



React Basics

Beginner • 4 weeks

100% completed

Continue learning

[View certificate](#) 🎓

Remove



Advanced JavaScript

Intermediate • 6 weeks

67% completed

[Continue learning](#)

Remove



Full-Stack Fundamentals

Intermediate • 8 weeks

5% completed

Continue learning

Remove

Contact Page

EQ

EduQuest

Explore • Learn • Thrive

Home

About

Courses

Blog

Team

Testimonials

Enrolled

Contact

Hi, abc

Logout

Contact Us

Have questions about our courses, pricing, or need technical support? We're here to help.

At EduQuest, we value our community of learners and educators. Whether you are looking for guidance on which course to take, need assistance with your account, or want to explore partnership opportunities, our team is ready to assist you.

We believe in prompt and personal communication. Once you submit your inquiry, it is routed directly to our support specialists who will ensure you get the answers you need to keep moving forward in your learning journey.

Get in touch directly

Email: support@eduquest.com

Response time: Typically within 24 hours

Name

Email

Message

Send message