Department of Computer Science and Engineering

NALANDA INSTITUTE OF TECHNOLOGY, CHANDAKA, BHUBANESWAR

(Affiliated to Biju Patnaik University of Technology, Odisha)

Project Report

ON

STUDYNOTION

SUBMITTED IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE of BACHLOR OF TECHNOLOHY in Computer Science and Engineering

Under the Guidance of

Prof. Gyana Prakash Bhuyan

UNDER BIJU PATNAIK UNIVERSITY OF TECHNOLOGY (BPUT)

SUBMITTED BY:

AJIT KUMAR GANDHI	[2001297101]
ABHIJIT DAS	[2001297095]
JAGNESWAR BEHERA	[2001297129]
BHARAT BHUSAN BEHERA	[2001297111]
CHANDRA KANTA MAHALIK	[2001297120]

PROJECT REPORT

ON

STUDYNOTION

SUBMITTED BY:

Ajit Kumar Gandhi 2001297101

Abhijit Das 2001297095

Jagneswar Behera 2001297129

Bharat Bhusan Behera 2001297111

Chandrakanta mahalik 2001297120

In partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering

Under the Guidance of

Prof. Gyana Prakash Bhuyan

NALANDA INSTITUTE OF TECHNOLOGY,

CHANDAKA, BHUBANESWAR-754005

2023-24

Department of Computer Science and Engineering

NALANDA INSTITUTE OF TECHNOLOGY, CHANDAKA, BHUBANESWAR-754005

2023-24

CERTIFICATE

This is to certify that, this report **"STUDYNOTION"** is bonafide work done **by Ajit kumar Gandhi(2001297101), Abhijit Das(2001297095), Jagneswar Behera (2001297129), Bharat Bhusan Behera(2001297111), Chandra kanta Mahalik(2001297120)** during 8th semester in partial fulfilment of the award of Bachelors Degree of Technology **in Computer Science and Engineering** under **Biju Patnaik University of Technology, Rourkela, Odisha.**

Project Guide **External Guide** **HOD, DEPT. OF CSE**

DECLARATION

We hereby declare that the project work "**STUDYNOTION**" submitted to the Dept of **Computer science & Engineering**, **Nalanda Institute of Technology ,BBSR**, is a record of an original work done by us under the guidance of Prof. **Gyana Prakash Bhuyan** and this project work is submitted in the partial fulfilment of the requirements for the award of the Degree of B.Tech . The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

This is to certify that the above declaration is true.

Name of the students

Ajit Kumar Gandhi **2001297101**

Abhijit Das **2001297095**

Jagneswar behera **2001297129**

Bharat Bhusan Behera **2001297111**

Chandrakanta Mahalik **200129712**

A C K N O W L E D GE M E N T

We are very grateful, thankful and wish to record my indebt ness to Prof. Gyana Prakash Bhuyan of Nalanda Institute of Technology, Bhubaneswar, for his active guidance and interest in this project work.

We again like to thank Prof. Narottam Sahu, H.O.D, Department of Computer Science and Engineering and for his continued drive for better quality in everything that allowed us to carry out our project work.

Lastly, word run to express our gratitude to my Parents and all the Professors, Lecturers, Technical and Official staffs and friends for their co-operation, constructive criticism and valuable suggestions during the preparation of project report.

Ajit Kumar Gandhi **[2001297101]**

Abhijit Das **[2001297095]**

Jagneswar Behera **[2001297129]**

Bharat Bhusan Behera **[2001297111]**

Chandra kanta mahalik **[2001297095]**

ABSTRACT

STUDYNOTION, specializing in ED-Tech education based platform. The website's objective is to provide high-quality compressor components with a user-friendly interface and excellent educational knowledge. Developed using HTML, CSS, JavaScript, ReactJS, Tailwind, Node.JS, and MongoDB, the site ensures a modern design, responsiveness, and easy navigation. JavaScript frameworks like React empower the creation of interactive web applications, while Material UI offers pre-designed components for customization and seamless integration. Email.js facilitates email sending without a server-side setup. The report includes CSS styling for the Home and About Us pages, focusing on responsive card designs and navigation elements. ReactJS provides the function of the website and Tailwind provides the style of the website.

INDEX

Sl.no	Content	
1.	Introduction	1-2
2.	Objective	3
3.	Technological Stack	4-9
4.	Step to create a website	10
5.	Component	11-24
6.	Product Description	25-26
7.	User friendly Interface	27-31
8.	System Architecture	32-36
9.	User Benefits	37
10.	Coding	38-45
11.	Conclusion	46
12.	Refences	47

INTRODUCTION

StudyNotion is a fully functional ed-tech platform that enables users to create, consume, and rate educational content. The platform is built using the MERN stack, which includes ReactJS, NodeJS, MongoDB, and ExpressJS.

This platform Provides most demanding skillset using IT Industry.

It is a easily understandable and comfortable platform for learners.

The front end of the platform is built using ReactJS, which is a popular JavaScript library for building user interfaces.

The back end of the platform is built using NodeJS and ExpressJS.

The database for the platform is built using MongoDB.

In the following sections, we will cover the technical details of the platform, including:

1. System architecture: The high-level overview of the platform's components and diagrams of the architecture.
2. Front-end: The description of the front-end architecture, user interface design, features, and functionalities of the front-end, and frameworks, libraries, and tools used.
3. Back-end: The description of the back-end architecture, features and functionalities of the back-end, frameworks, libraries, tools used, and data models and database schema.
4. API Design: The description of the API design, list of API endpoints, their functionalities, and sample API requests and responses.
5. Deployment: The description of the deployment process, hosting environment and infrastructure, and deployment scripts and configuration.
6. Testing: The description of the testing process, types of testing, test frameworks and tools used.
7. Future Enhancements: The list of potential future enhancements to the platform, explanation of how these enhancements would improve the

platform, estimated timeline and priority for implementing these enhancements.

In summary, StudyNotion is a versatile and intuitive ed-tech platform that is designed to provide an immersive learning experience to students and a platform for instructors to showcase their expertise. In the following sections, we will delve into the technical details of the platform, which will provide a comprehensive understanding of the platform's features and functionalities.

OBJECTIVE

StudyNotion is a versatile and intuitive ed-tech platform that is designed to provide an immersive learning experience to students and a platform for instructors to showcase their expertise. In the following sections, we will delve into the technical details of the platform, which will provide a comprehensive understanding of the platform's features and functionalities.

The website's objective is to offer an intuitive and user-friendly interface that enables customers to quickly find and purchase the Course they need. It provides technical skill and advice to help to added extra technical skill for help in future.

Product Accessibility: A learner easily come to the website and go to the dashboard and access their course.

Reliability and Efficiency: A learner show their course progress of their account. Before add any video a mail comes to the learners mail box so that they know and watch the new video

Customer Convenience: To provide an intuitive website interface and navigation system, enabling customers to quickly locate and purchase the required course and start learning from beginning.

TECHNOLOGY STACK

HTML, CSS, JavaScript, ReactJS, Tailwind, NodeJS, and MongoDB, facilitating the development of dynamic and responsive web interfaces .This powerful combination ensures a user-centric experience and modern design elements, enhancing the overall functionality and visual appeal of our project.

1. **HTML (Hypertext Mark-up Language):**
    - **Semantic Structure:** HTML provides a structured way to organize content, enhancing accessibility and SEO by clearly defining the purpose of each element.
    - **Accessibility:** Properly structured HTML ensures compatibility with assistive technologies, improving accessibility for users with disabilities.
    - **Foundation for Web Development:** HTML serves as the backbone of web development, laying the groundwork for styling and interactivity.

2. **CSS (Cascading Style Sheets):**
    - **Design Consistency:** CSS allows developers to create consistent designs across multiple pages, ensuring a unified brand experience.
    - **Responsive Design:** CSS media queries enable websites to adapt to various screen sizes and devices, enhancing usability on mobile and desktop platforms.
    - **Styling Flexibility:** It is used to control the layout, colors, fonts, and other visual aspects of the website.

3. **Tailwind:**
    - **Styling:** Tailwind is a framework of CSS. Tailwind CSS works smoothly with a plethora of frameworks like Next, React, Angular, and more – and even our OG HTML.

- **Cross-Browser Compatibility**: Tailwind ensures consistent rendering
- across different browsers, reducing the need for extensive browserspecific CSS fixes

4. JavaScript:

- **Versatile Language:** JavaScript is a versatile programming language capable of running both on the client-side (browser) and server-side (Node.js), making it suitable for full-stack development.
- JavaScript's versatility and ubiquity make it an essential skill for developers across various domains, offering opportunities for building diverse applications.
- **Dynamic and Interactivity:** JavaScript enables the creation of dynamic and interactive web pages by allowing developers to manipulate DOM elements, respond to user actions, and modify page content in real-time.
- It supports event-driven programming, enabling the execution of code in response to user interactions like clicks, scrolls, and keystrokes, enhancing user engagement and interactivity.
- Popular libraries and frameworks like React, Vue.js, and AngularJS provide developers with powerful abstractions and tools for building scalable and maintainable web applications.
- Additionally, JavaScript has a rich ecosystem of package managers (npm, Yarn), build tools (Web pack, Babel), and testing frameworks (Jest, Mocha), empowering developers to streamline development workflows and build high-quality software efficiently.

5. **ReactJs:**
    - ReactJs is a JavaScript library for building user interfaces, particularly for single-page applications. It was developed by Facebook and is now maintained by Facebook and a community of individual developers and companies.
    - ReactJs allows developers to build reusable UI components, which can be combined to create complex user interfaces. This modular approach to UI development can make code more maintainable and easier to debug.
    - ReactJs uses a virtual DOM to improve performance by minimizing direct manipulation of the DOM. When a component's state changes, ReactJs update the virtual DOM and then efficiently update the actual DOM, resulting in faster rendering and a smoother user experience.

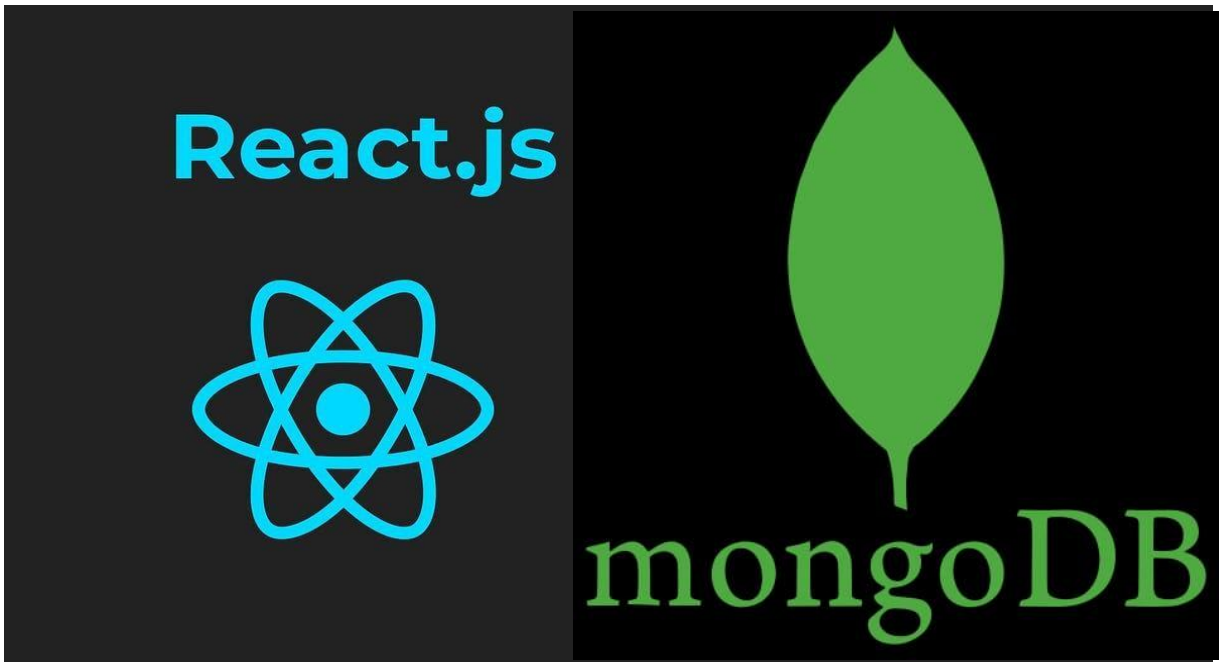
6. **Node.JS:**
    - Node.js (Node js) is an open-source and cross-platform JavaScript runtime environment. It runs on Chrome's V8 JavaScript engine. It allows developers to run JavaScript code on the server. Node.js allows developers comfortable with this versatile language to get into the server-side world.
    - Node.js has a unique advantage because millions of frontend developers who write JavaScript for the browser can now write server-side code without needing to learn a completely new language.
    - Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. Node.js is able to leverage the work of the engineers that made (and will continue to make) the Chrome JavaScript runtime blazing fast, and this allows Node.js to benefit from the huge performance improvements and the Just-In-Time compilation that V8 performs. Thanks to this, JavaScript code running in Node.js can become very performant.

- A Node.js app is run by a single process, without creating a new thread for every request. Node provides a set of asynchronous I/O primitives in its standard library that will prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making a blocking behavior an exception rather than the normal

7. MongoDB:

- MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need
- MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.
- If you have huge amount of data to be stored in tables, think of MongoDB before RDBMS databases. MongoDB has built-in solution for partitioning and sharding your database.

HTML
CSS
JS


React.js
mongoDB

STEP TO CREATE A WEBSITE

Creating a website involves several steps, here's a simplified breakdown:

Planning: Define the purpose of your website, target audience, and key features. Outline the site structure, content, and design preferences.

Choose a Domain Name: Select a unique and relevant domain name that reflects your brand or website's purpose. Register it through a domain registrar.

Select a Hosting Provider: Choose a hosting service that suits your website's needs (shared, VPS, dedicated, etc.). Ensure it aligns with your expected traffic and technical requirements.

Select a Website Building Platform: Choose a platform (WordPress, Wix, Square space, etc.) based on your technical skills, customization needs, and desired functionalities.

Design Your Website: Use website builders or templates to create the layout, design elements, and user interface. Ensure its mobile-responsive for different devices.

Content Creation: Develop and organize compelling content (text, images, and videos) that aligns with your brand and engages your audience.

SEO and Optimization: Implement SEO strategies to enhance visibility. Optimize content, Meta tags, images, and site speed for better search engine rankings.

Testing: Thoroughly test your website across different browsers, devices, and screen sizes to ensure functionality and user-friendliness.

Deployment: Once everything is set, launch your website. Promote it through various channels to attract visitors.

Maintenance and Updates: Regularly update content, fix bugs, and implement security measures. Monitor site performance and make necessary improvements.

COMPONENT

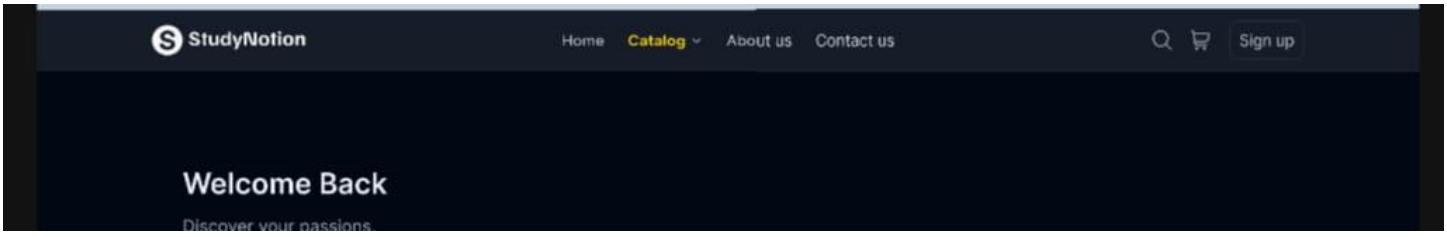
5.1 Navbar:

A navbar, short for navigation bar, is a website feature that contains links to various sections or pages of the site. It is typically located at the top of the page and provides users with an easy way to navigate the site.

In the context of the website Navbar, it contains the following links:

- **Home:** This is typically the landing page of the website, which provides an overview of the company and its services. It is the first page that users see when they visit the site.
- **Catalogue:** Catalogue is used for describe all the courses of the website. A learner simply find and show all the courses inside catalogue.
- **About Us:** This link leads to a page that provides information about the company, such as its history, mission, and values. It may also include details about the team, including their experience and expertise.
- **Contact Us:** This link leads to a page that provides users with various ways to get in touch with the company, such as by phone, email, or contact form. It may also include the company's physical address and operating hours.

Navbar Screenshot:



5.2 Home page

The home page of STUDYNOTION provides a comprehensive overview of the company's offerings and contact information. Here is a detailed description of the content on the home page:

Website Name and Focus: The company is named STUDYNOTION, highlighting its courses and video content.

Main Menu: The menu options include "Home," "About Us,"Catalogue," and "Contact Us," offering easy navigation to different sections of the website.

Website Offering: This website offers web developement , C++ language, Java, Python and AI .

Contact Information: All the contact information and courses information is given inside the contact us tag.

Quick Links: The quick links section likely directs users to important pages such as Home, About Us, Contact Us, FAQs, and the physical location on a map.

Home page screenshot:

5.3 Footer:

Footer is a section that appears at the bottom of a website, typically spanning the width of the page. It usually contains information that is relevant to the website as a whole, rather than a specific page or section.

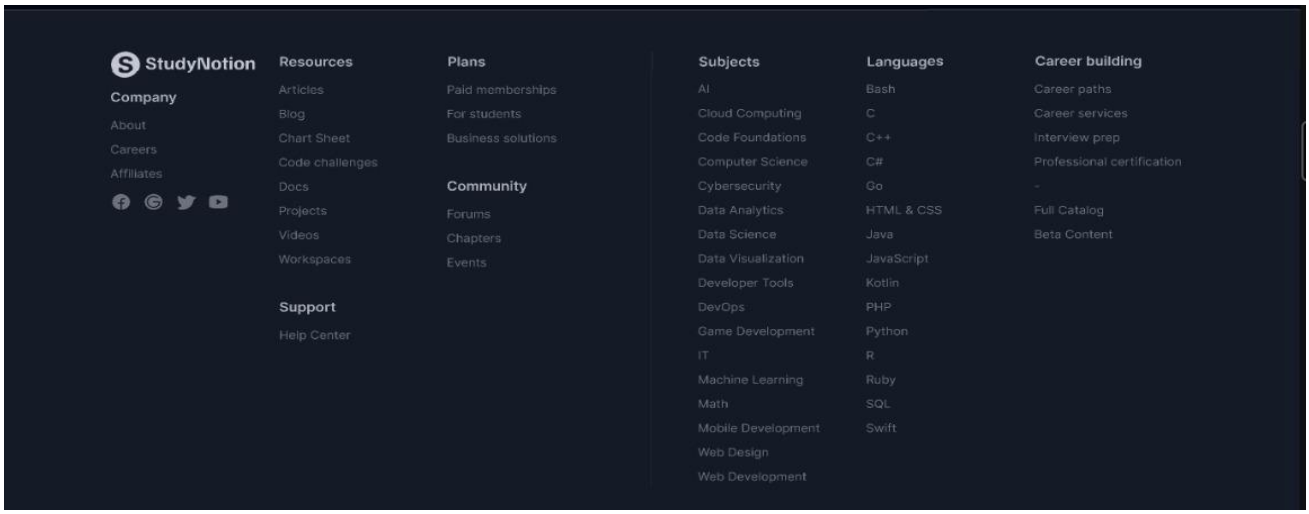
Copyright information: This typically includes the copyright symbol, the year, and the name of the website or company. It may also include a statement regarding the website's terms of use or privacy policy.

Contact information: This may include the company's physical address, phone number, and email address. Some websites may also include a contact form or a link to a contact page.

Social media links: Many websites include links to their social media profiles in the footer, making it easy for users to connect with them on platforms like Facebook, Twitter, and Instagram.

Navigation links: Some websites include additional navigation links in the footer, such as links to legal pages (e.g. terms of service, privacy policy), a site map, or a search bar.

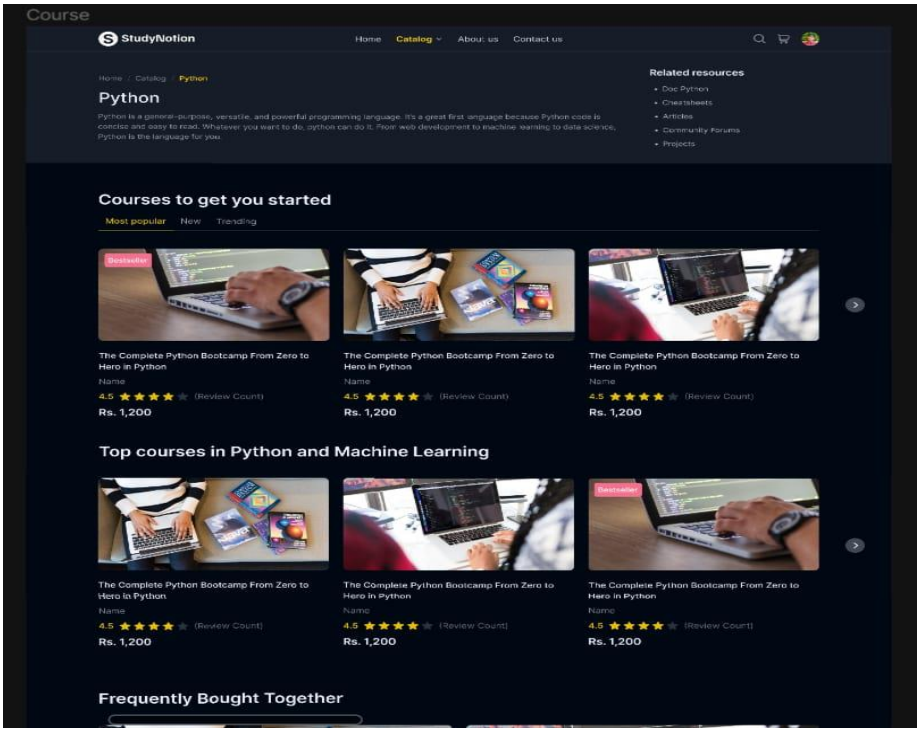
Footer Screenshot:

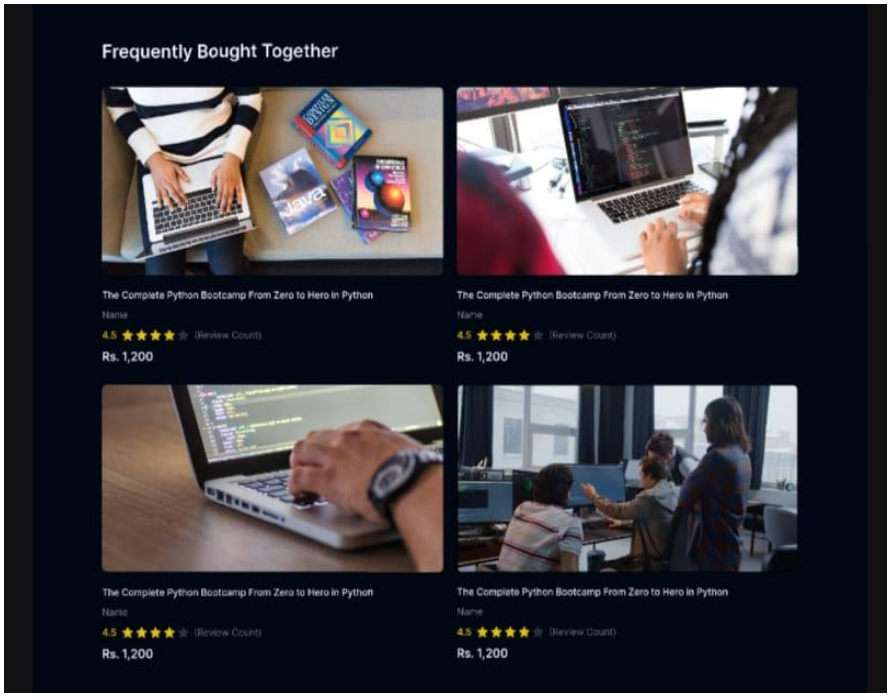


5.4 catalogue:

- Catalogue provides all the courses option , rating , review and all the courses description.
- If anyone can purchase the course simply click on catalogue and purchase their required course .
- All the purchase amount and course rating is given in catalog.

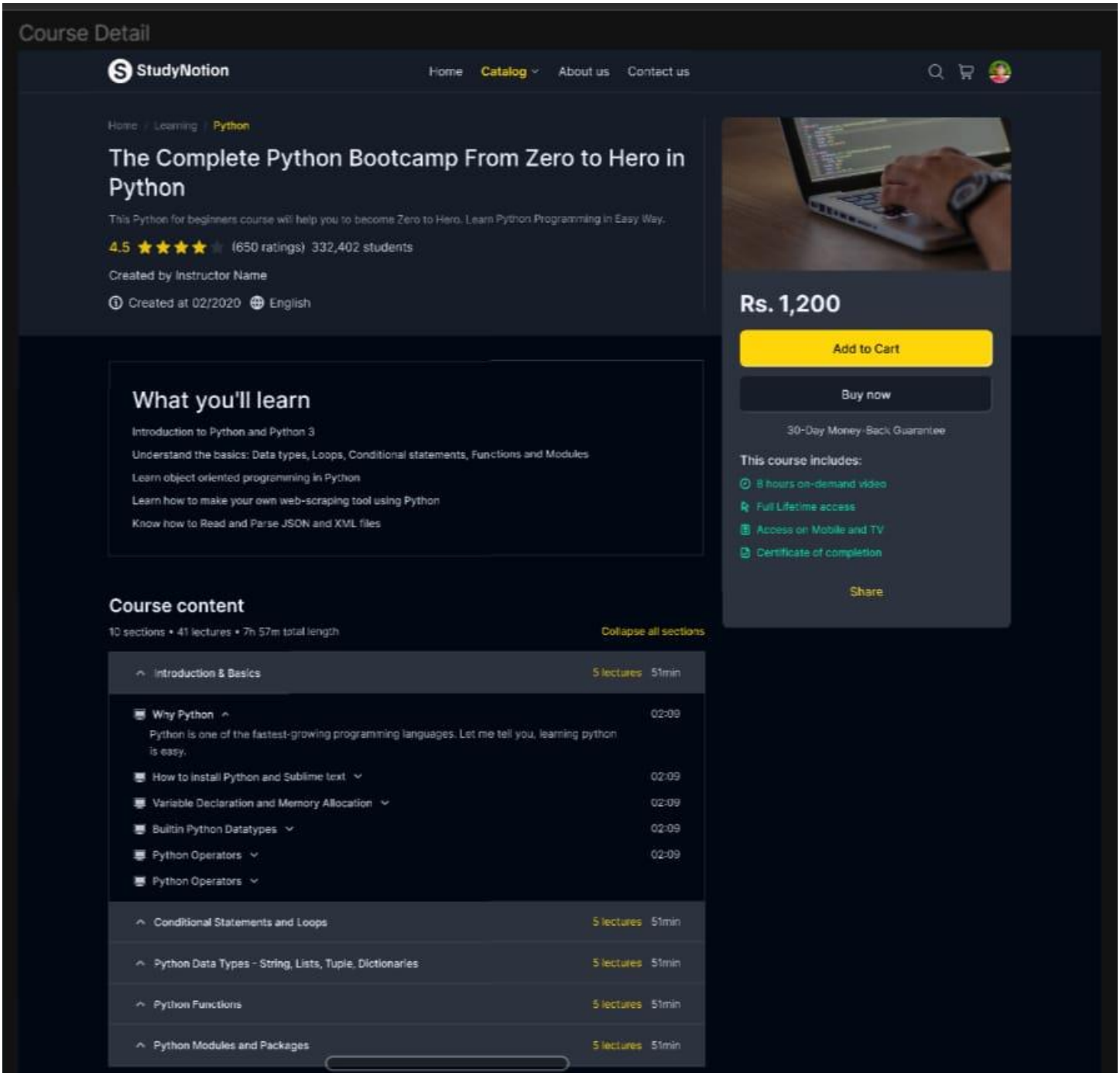
Catalogue Screenshot:





Course Purchase links:

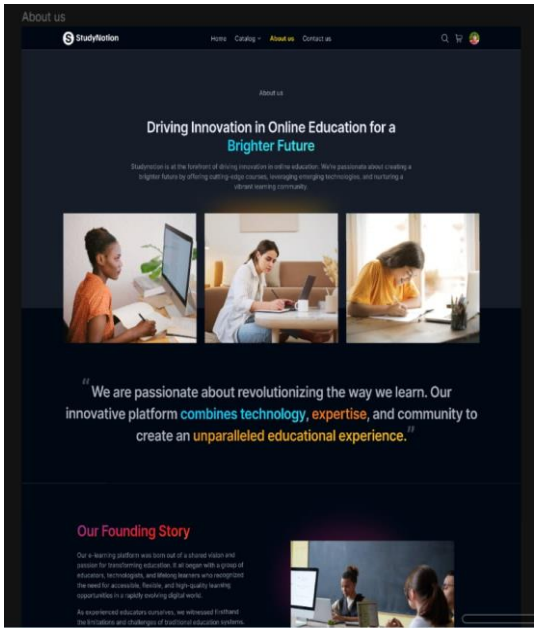
Here every card have links to their respective courses with details.

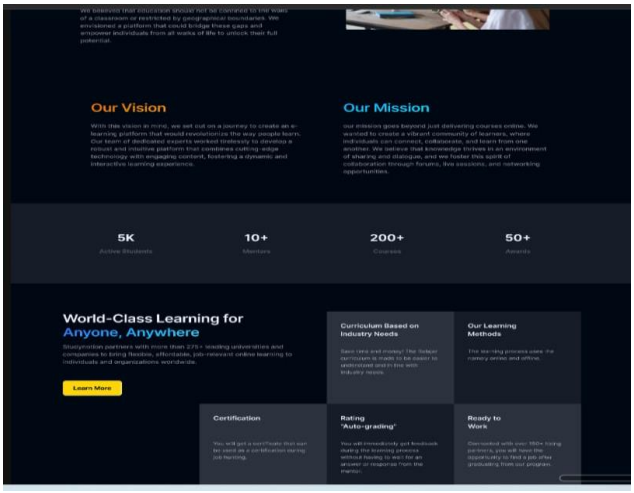


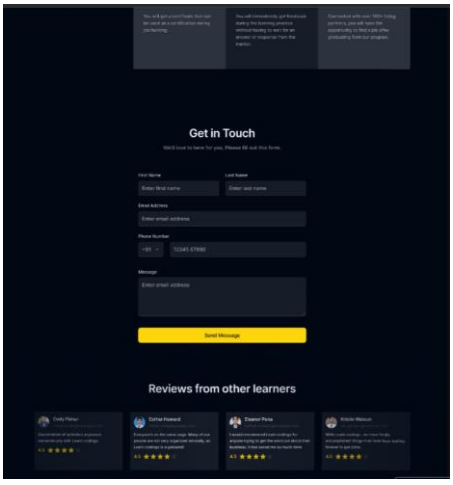
5.5 About Us:

An "About Us" page provides visitors with essential information about the company. It includes a company overview, mission and values statement, team member bios, company culture insights, achievements and awards, customer testimonials, community involvement details, and clear contact information, facilitating communication with visitors.

About Us Screenshot:





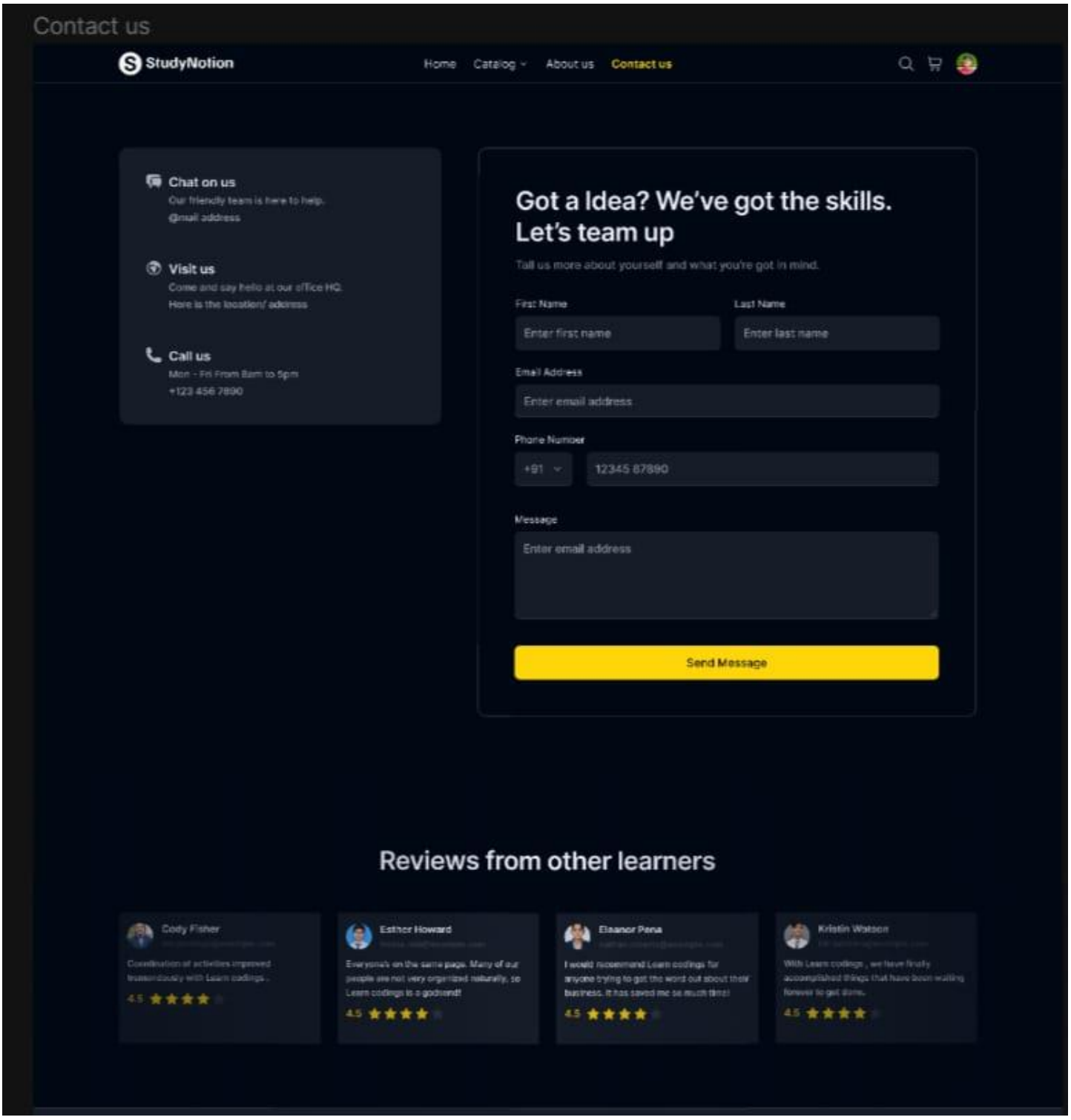


5.6 Contact Us:

The "Contact Us" section of a website is crucial for facilitating communication between visitors and the company. It typically includes various means through which visitors can reach out to the company, such as:

- **Contact Form:** A form that visitors can fill out with their name, email address, subject, and message. This form allows for direct communication and enables the company to respond to inquiries efficiently.
- **Email Address:** A dedicated email address where visitors can send inquiries, questions, or feedback. Providing an email address allows for more personalized communication and ensures that messages are delivered directly to the appropriate department or individual.
- **Phone Number:** A contact number that visitors can call for immediate assistance or inquiries. Including a phone number adds a sense of accessibility and allows visitors to speak directly with a representative if they prefer.
- **Social Media Links:** Links to the company's official social media profiles, such as Facebook, Twitter, LinkedIn, etc. Visitors can use these platforms to reach out to the company, ask questions, or provide feedback in a more informal setting.

Contact Us Screenshot: (Desktop View)



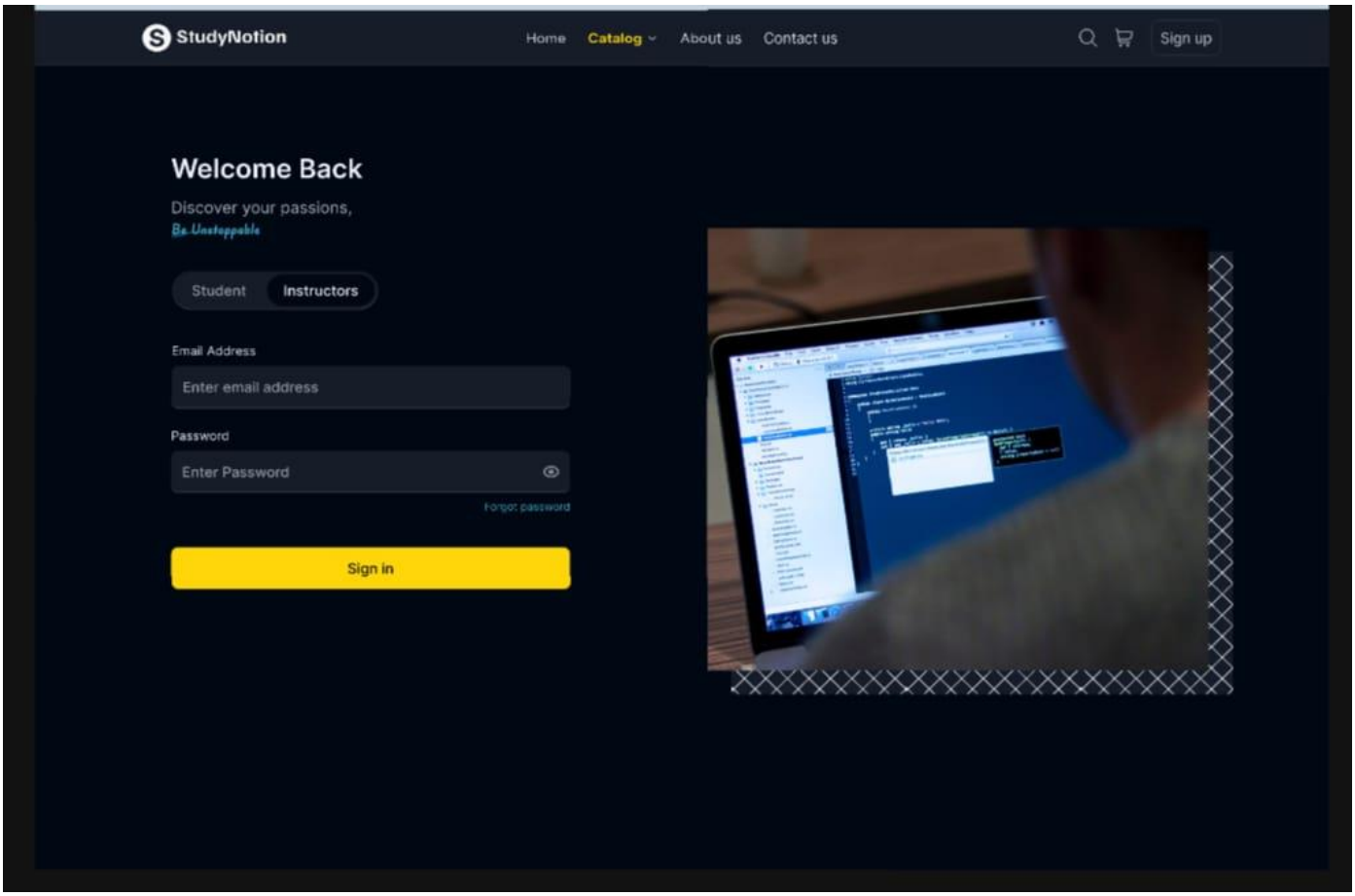
5.7 Login:

The "Login" section provides existing users with a way to access their accounts by entering their credentials, typically a username or email address and a password.

Users are usually directed to this section when they want to access restricted areas of the website, such as their account dashboard, personalized settings, or to complete a transaction.

Once logged in, users may have access to additional features, such as saved preferences, order history, or the ability to manage their account details.

Login Screenshot: (Desktop View)



5.7 Sign Up:

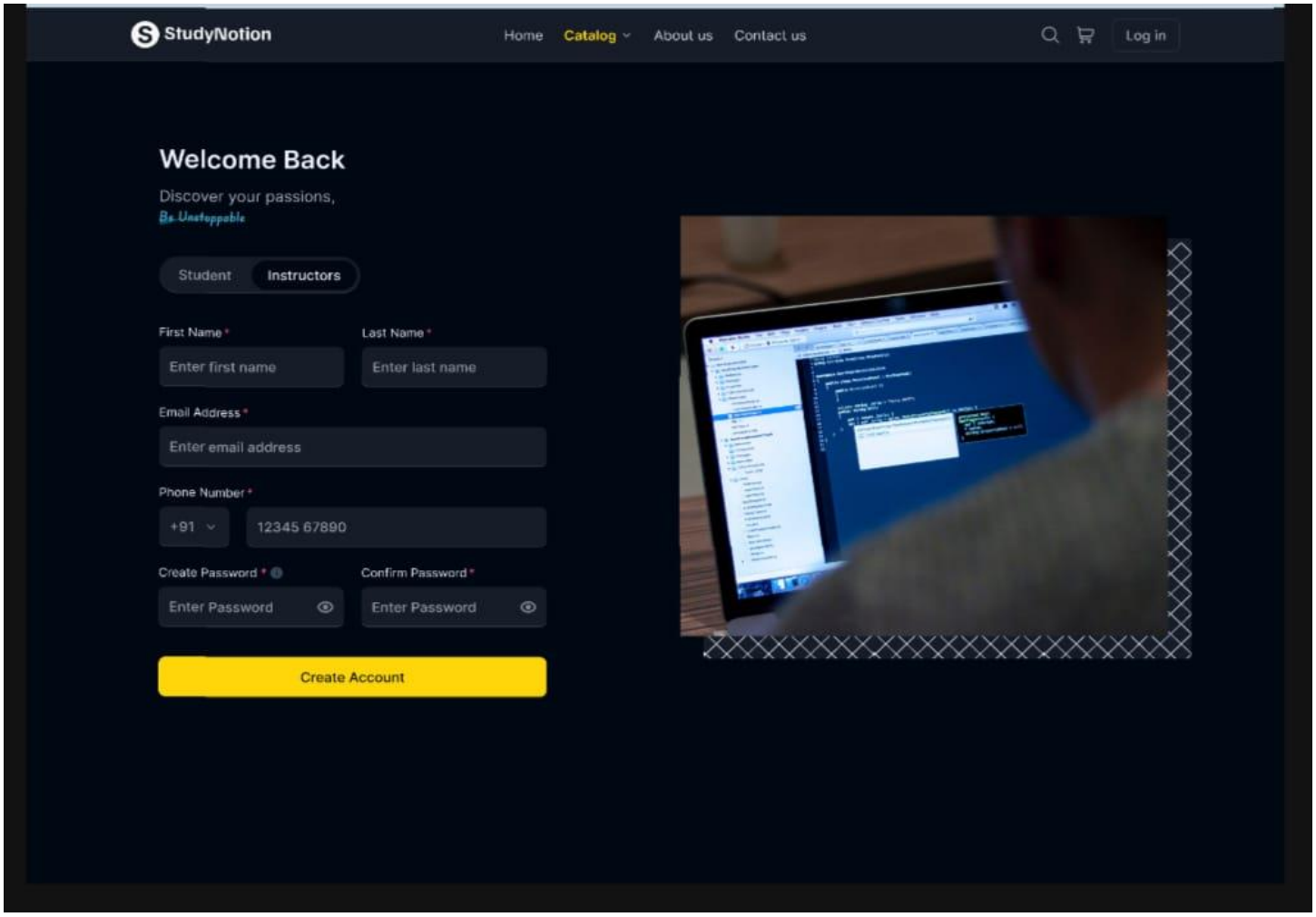
The "Sign Up" section allows new users to create an account on the website by providing necessary information, such as their name, email address, and creating a password.

Users may be prompted to sign up when they want to access exclusive content, make purchases, or subscribe to newsletters or updates.

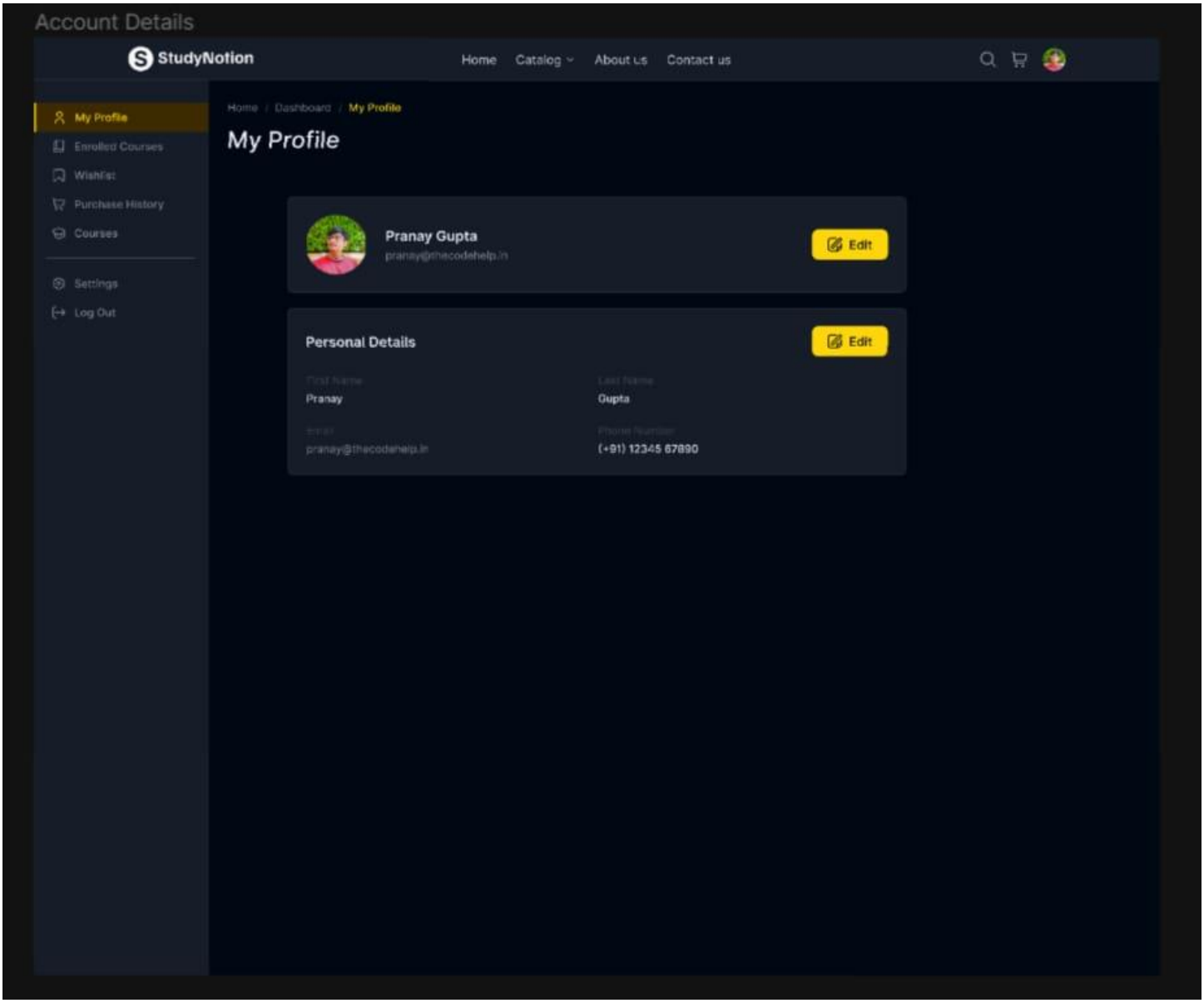
After completing the sign-up process, users typically receive a confirmation email to verify their email address and activate their account.

Once signed up, users can then proceed to log in to their newly created account using the credentials they provided during the sign-up process.

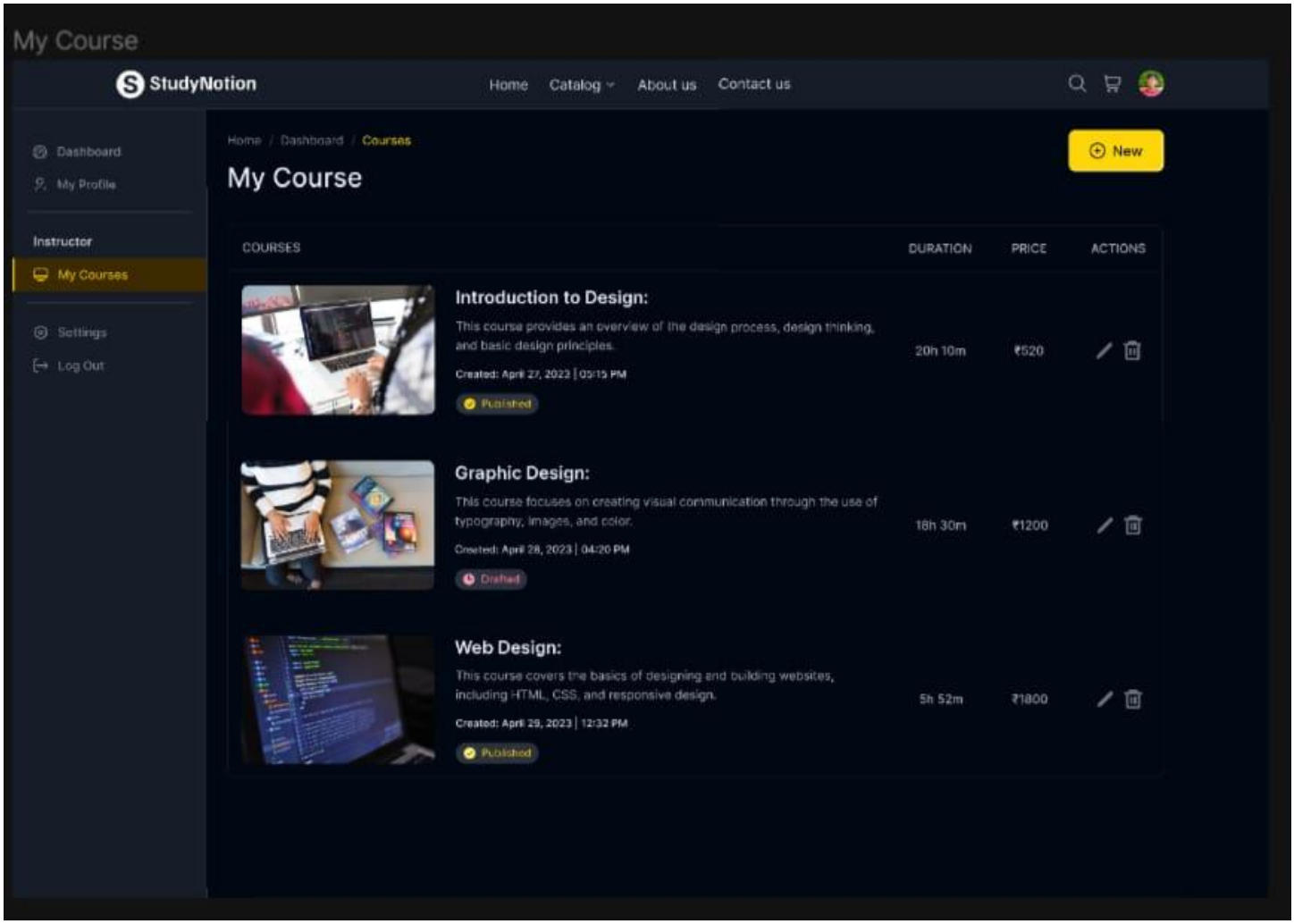
Sign Up Screenshot:



5.8 Student Dashboard View:



5.9 Instructor Dashboard view:



PRODUCT DESCRIPTION

StudyNotion is a fully functional ed-tech platform that enables users to create, consume, and rate educational content. The platform is built using the MERN stack, which includes ReactJS, NodeJS, MongoDB, and ExpressJS.

StudyNotion aims to provide:

- A seamless and interactive learning experience for students, making education more accessible and engaging.
- A platform for instructors to showcase their expertise and connect with learners across the globe.

In the following sections, we will cover the technical details of the platform, including:

1. System architecture: The high-level overview of the platform's components and diagrams of the architecture.
2. Front-end: The description of the front-end architecture, user interface design, features, and functionalities of the front-end, and frameworks, libraries, and tools used.
3. Back-end: The description of the back-end architecture, features and functionalities of the back-end, frameworks, libraries, tools used, and data models and database schema.
4. API Design: The description of the API design, list of API endpoints, their functionalities, and sample API requests and responses.
5. Deployment: The description of the deployment process, hosting environment and infrastructure, and deployment scripts and configuration.
6. Testing: The description of the testing process, types of testing, test frameworks and tools used.
7. Future Enhancements: The list of potential future enhancements to the platform, explanation of how these enhancements would improve the

platform, estimated timeline and priority for implementing these enhancements.

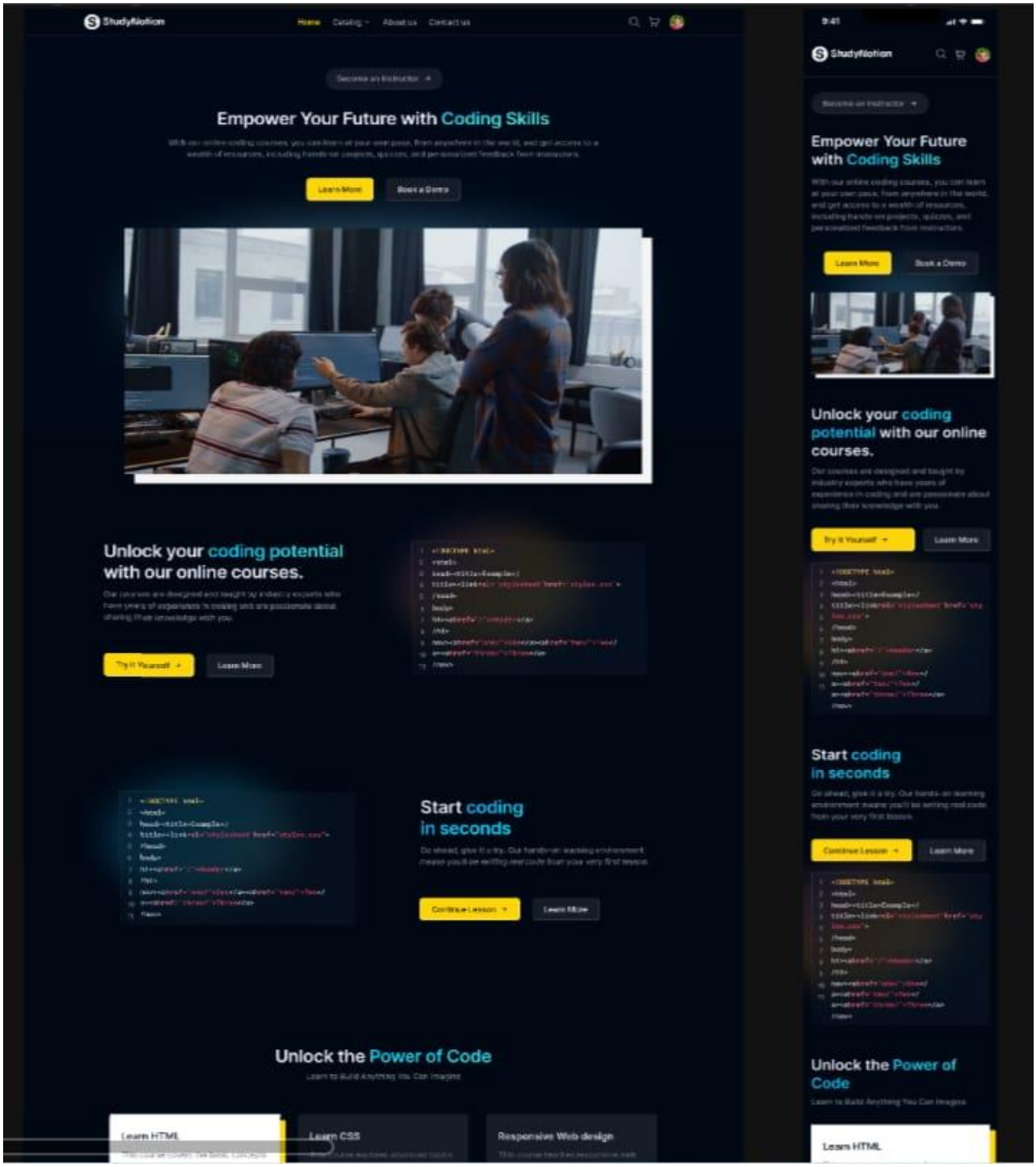
In summary, StudyNotion is a versatile and intuitive ed-tech platform that is designed to provide an immersive learning experience to students and a platform for instructors to showcase their expertise. In the following sections, we will delve into the technical details of the platform, which will provide a comprehensive understanding of the platform's features and functionalities.

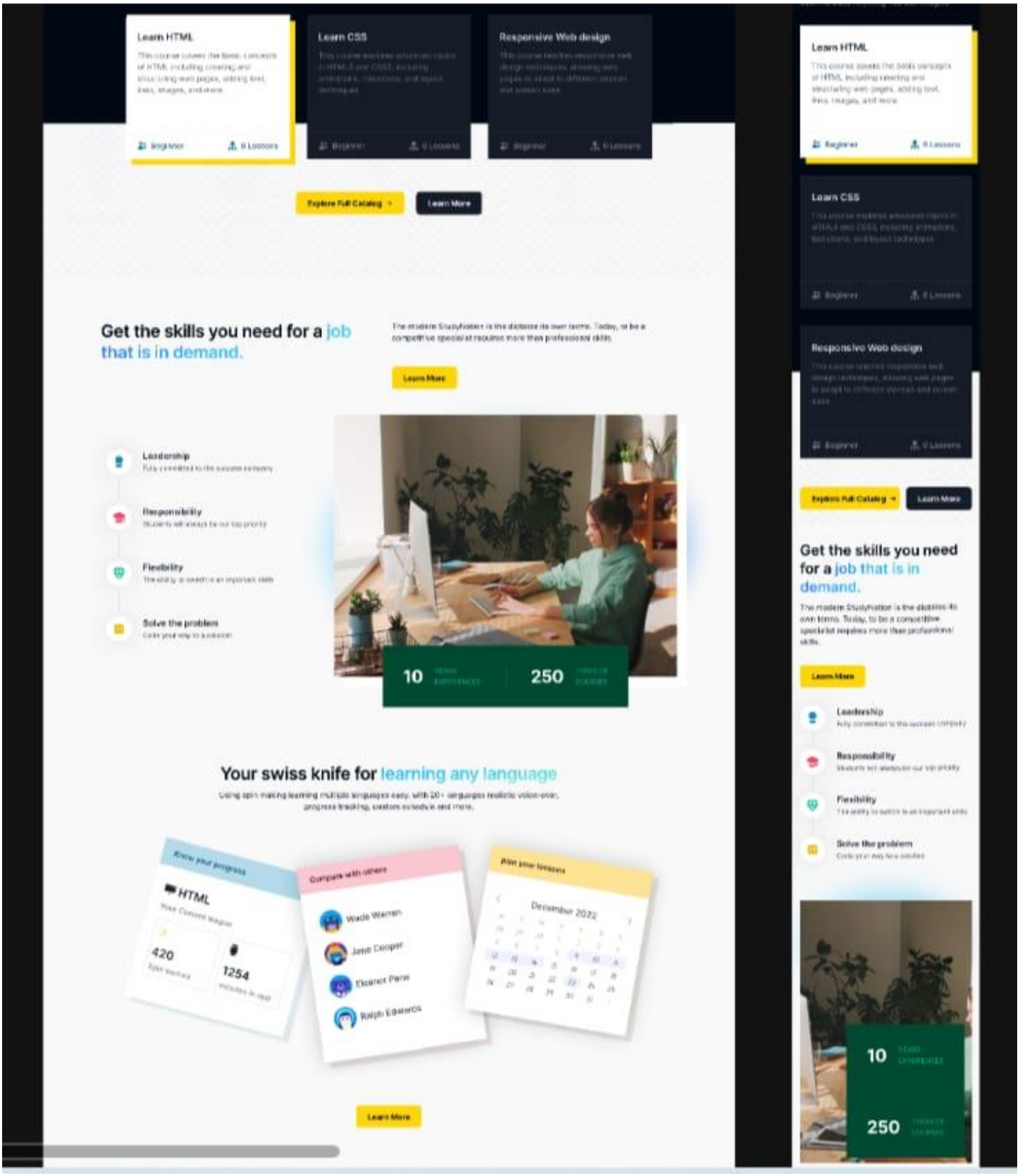
USER FRIENDLY INTERFACE

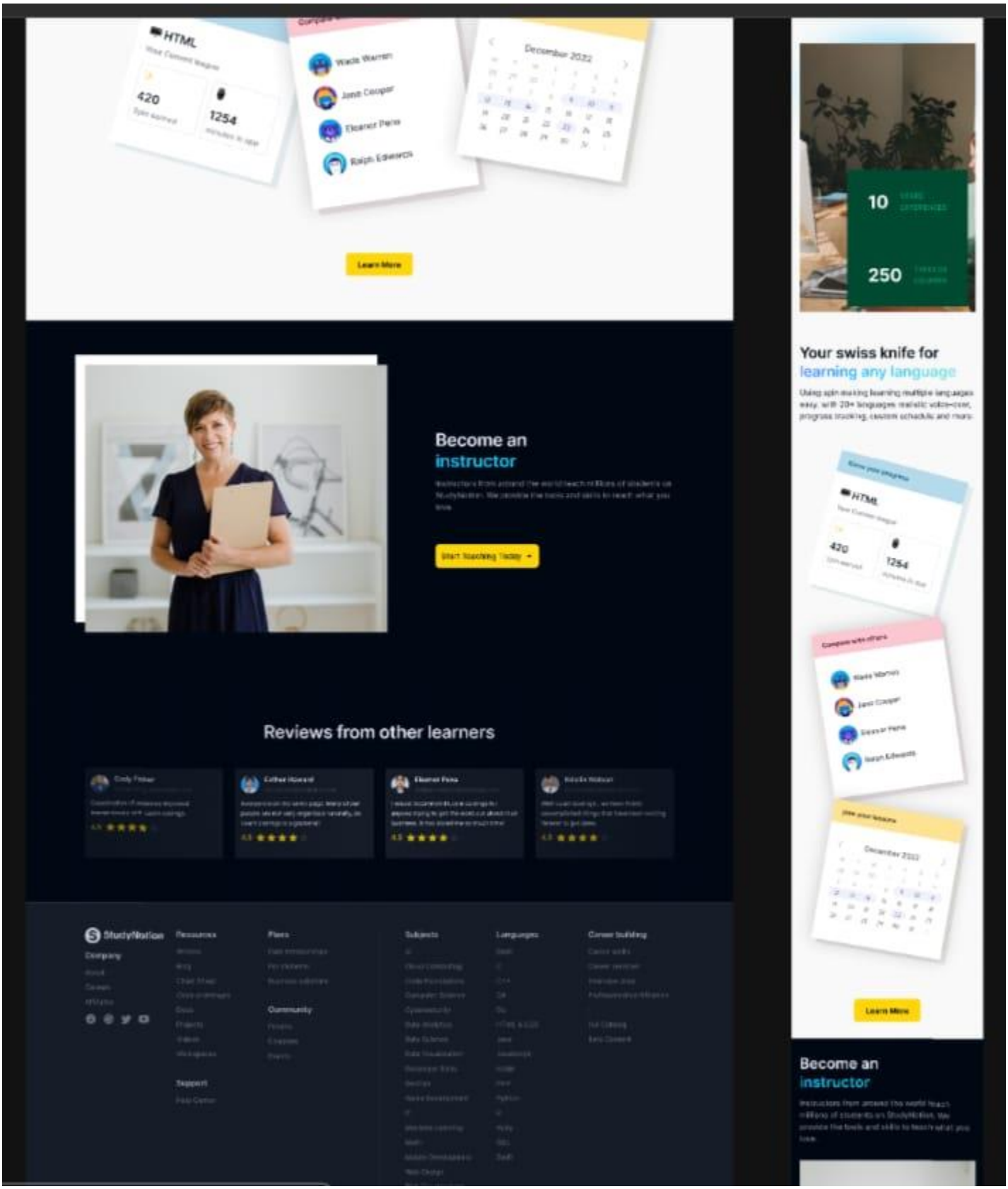
A user-friendly interface is a key component of any successful website or application, as it greatly impacts the overall user experience.

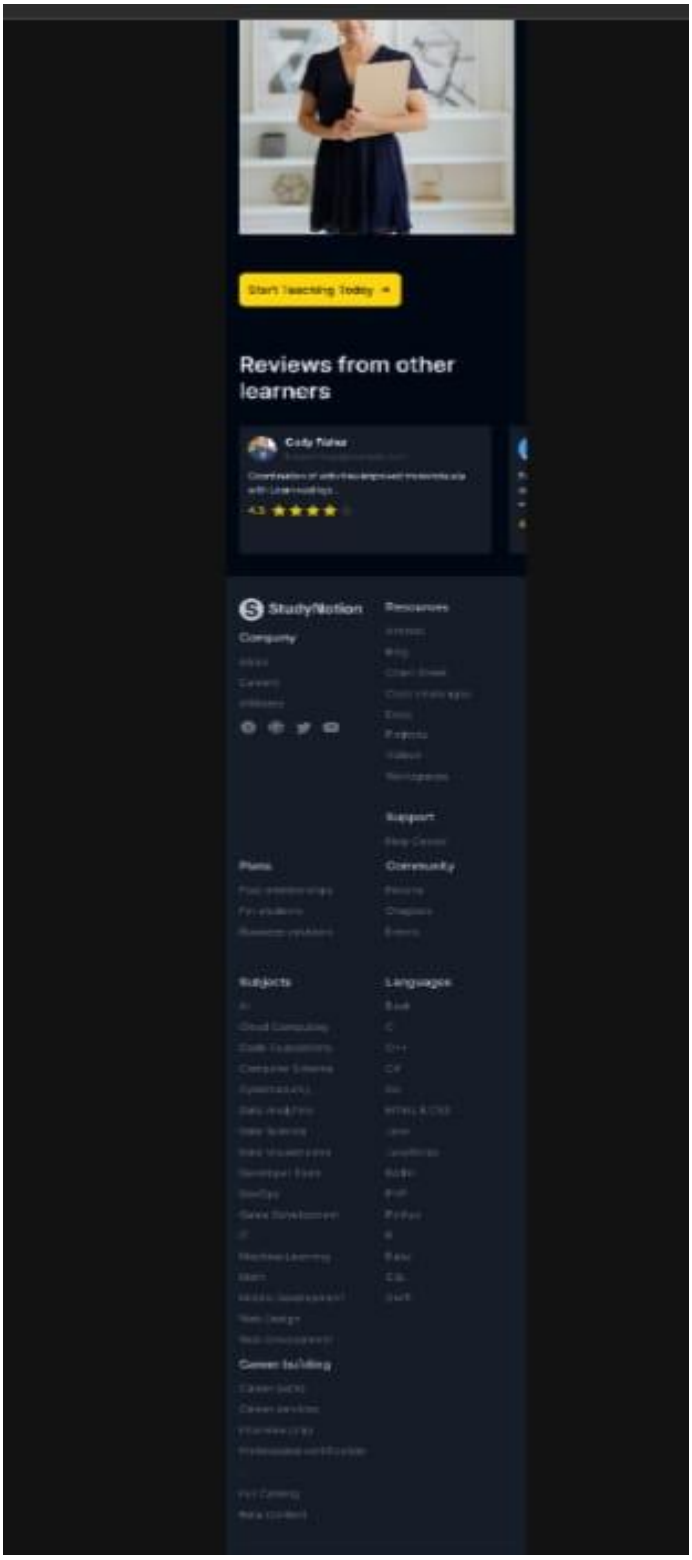
- **Intuitive Navigation:** The interface should be easy to navigate, with clear menus, buttons, and links that guide users seamlessly through the website or app. Users should be able to find what they're looking for quickly and without confusion.
- **Clear Layout and Organization:** Information should be presented in a logical and organized manner, with consistent layouts and design elements throughout the interface. This helps users understand the structure of the content and find what they need more efficiently.
- **Responsive Design:** The interface should adapt to different screen sizes and devices, ensuring a consistent and optimized experience for users across desktops, laptops, tablets, and smartphones.
- **Readable Typography:** Text should be easy to read, with appropriate font sizes, styles, and spacing. Important information should be highlighted effectively, and paragraphs should be broken up into digestible chunks for improved readability.
- **Visual Hierarchy:** Elements such as headings, subheadings, and images should be used to create a clear visual hierarchy that guides users' attention and helps them prioritize information on the page.
- **Minimalistic Design**: Avoid clutter and unnecessary distractions by keeping the interface clean and minimalistic. Use whitespace effectively to give content room to breathe and focus users' attention on the most important elements.

Responsive Designs Screenshot:




Learn HTML
Learn CSS
Responsive Web design
Explore Full Catalog
Learn More
Get the skills you need for a job that is in demand.
Learn More
Leadership
Responsibility
Flexibility
Solve the problem
10
250
Your swiss knife for learning any language
Learn More


HTML
420
1254
Wade Warren
Jane Cooper
Eleanor Pena
Ralph Edwards
December 2022
Learn More
10
250
Become an
instructor
Start Teaching Today
Reviews from other learners
Your swiss knife for
learning any language
Learn More
Become an
instructor


Start Teaching Today
Reviews from other learners
StudyNotion

System Architecture

The StudyNotion ed-tech platform consists of three main components: the front end, the back end, and the database. The platform follows a client-server architecture, with the front end serving as the client and the back end and database serving as the server.

Front-end

The front end of the platform is built using ReactJS, which is a popular JavaScript library for building user interfaces. ReactJS allows for the creation of dynamic and responsive user interfaces, which are critical for providing an engaging learning experience to the students. The front end communicates with the back end using RESTful API calls.

Back-end

The back end of the platform is built using NodeJS and ExpressJS, which are popular frameworks for building scalable and robust server-side applications. The back end provides APIs for the front end to consume, which include functionalities such as user authentication, course creation, and course consumption. The back end also handles the logic for processing and storing the course content and user data.

Database

The database for the platform is built using MongoDB, which is a NoSQL database that provides a flexible and scalable data storage solution. MongoDB allows for the storage of unstructured and semi-structured data, which is useful for storing course content such as videos, images, and PDFs. The database stores the course content, user data, and other relevant information related to the platform.

Front-end

The front end is part of the platform that the user interacts with. It's like the "face" of the platform that the user sees and interacts with. The front end of StudyNotion is designed using a tool called Figma, which is a popular design tool that allows for the creation of clean and minimal user interfaces. You can take a look at the Figma design for the StudyNotion front-end.

The front end of StudyNotion has all the necessary pages that an ed-tech platform should have. Some of these pages are:

For Students:

- Homepage: This page will have a brief introduction to the platform, as well as links to the course list and user details.
- Course List: This page will have a list of all the courses available on the platform, along with their descriptions and ratings.
- Wishlist: This page will display all the courses that a student has added to their wishlist.
- Cart Checkout: This page will allow the user to complete the course purchase.
- Course Content: This page will have the course content for a particular course, including videos, and other related material.
- User Details: This page will have details about the student's account, including their name, email, and other relevant information.
- User Edit Details: This page will allow the student to edit their account details.

For Instructors:

- Dashboard: This page will have an overview of the instructor's courses, as well as the ratings and feedback for each course.

- Insights: This page will have detailed insights into the instructor's courses, including the number of views, clicks, and other relevant metrics.
- Course Management Pages: These pages will allow the instructor to create, update, and delete courses, as well as manage the course content and pricing.
- View and Edit Profile Details: These pages will allow the instructor to view and edit their account details.

For Admin (this is for future scope)**:**

- Dashboard: This page will have an overview of the platform's courses, instructors, and students.
- Insights: This page will have detailed insights into the platform's metrics, including the number of registered users, courses, and revenue.
- Instructor Management: This page will allow the admin to manage the platform's instructors, including their account details, courses, and ratings.
- Other Relevant Pages: The admin will also have access to other relevant pages, such as user management and course management pages.

To build the front end, we use frameworks and libraries such as ReactJS, which is a popular JavaScript library for building user interfaces. We also use CSS and Tailwind, which are styling frameworks that help make the user interface look good and responsive. Additionally, we use some npm packages to add extra functionality to the front end. To manage the state of the application, we use Redux, which is a popular state management library for React. Finally, we use a development environment called VSCode, which is a popular code editor, to develop the front end.

Back-end

Description of the Back-end Architecture:

StudyNotion uses a monolithic architecture, with the backend built using Node.js and Express.js, and MongoDB as the primary database. Monolithic architecture refers to a design approach where all the modules of the application are combined into a single large program, with a single codebase, to enable better control, security, and performance.

Node.js is a popular JavaScript runtime that allows us to run JavaScript code outside of the browser. Express.js is a web application framework that simplifies the process of building web applications in Node.js. MongoDB is a popular NoSQL database that allows for flexible data storage and retrieval, making it a suitable choice for complex applications like StudyNotion.

Features and Functionalities of the Back-end:

The back end of StudyNotion provides a range of features and functionalities, including:

1. User authentication and authorization: Students and instructors can sign up and log in to the platform using their email addresses and password. The platform also supports OTP (One-Time Password) verification and forgot password functionality for added security.
2. Course management: Instructors can create, read, update, and delete courses, as well as manage course content and media. Students can view and rate courses.
3. Payment Integration: Students will purchase and enrol on courses by completing the checkout flow that is followed by Razorpay integration for payment handling.
4. Cloud-based media management: StudyNotion uses Cloudinary, a cloud-based media management service, to store and manage all media content, including images, videos, and documents.

5. Markdown formatting: Course content in document format is stored in Markdown format, which allows for easier display and rendering on the front end.

Frameworks, Libraries, and Tools used:

The back end of StudyNotion uses a range of frameworks, libraries, and tools to ensure its functionality and performance, including:

1. Node.js: Node.js is used as the primary framework for the back end.
2. MongoDB: MongoDB is used as the primary database, providing a flexible and scalable data storage solution.
3. Express.js: Express.js is used as a web application framework, providing a range of features and tools for building web applications.
4. JWT: JWT (JSON Web Tokens) are used for authentication and authorization, providing a secure and reliable way to manage user credentials.
5. Bcrypt: Bcrypt is used for password hashing, adding an extra layer of security to user data.
6. Mongoose: Mongoose is used as an Object Data Modeling (ODM) library, providing a way to interact with MongoDB using JavaScript.

USER BENIFITS

1. **Instructor:** In this website all the experienced and expertise mentor will be teach the courses . Teachers will be teach the content in Hinglish. This website will provided the code editor for practicing in future and provide so many quizzes related to the course for enhance the courses and skill perfectly.

2. **Quality Assurance:** In this website provides both live and recoded video, so that if learners faces any problem or doubt in live classes they can easily access the live classes in recoded format. All the recoded video provides in high quality. After learners fully finished the courses they can get course certificate.

3. **Progress bar:** All the learners can easily know how many videos they are completed in their course.

4. **Skill**: This website provides the content or courses based on software company requirement. The latest technology will be offered.

5. **Mode of education:** In this educational platform Provides online mode to learn. It is fully based on online.

6. **Responsive Customer Service:** Users can benefit from responsive customer service through various communication channels, such as phone and email.

CODING

```
import { useEffect } from "react"
import "./App.css"
// Redux
import { useDispatch, useSelector } from "react-redux"
// React Router
import { Route, Routes, useNavigate } from "react-router-dom"

// Components
import Navbar from "./components/Common/Navbar"
import OpenRoute from "./components/core/Auth/OpenRoute"
import PrivateRoute from "./components/core/Auth/PrivateRoute"
import AddCourse from "./components/core/Dashboard/AddCourse"
import Cart from "./components/core/Dashboard/Cart"
import EditCourse from "./components/core/Dashboard/EditCourse"
import EnrolledCourses from "./components/core/Dashboard/EnrolledCourses"
import Instructor from "./components/core/Dashboard/Instructor"
import MyCourses from "./components/core/Dashboard/MyCourses"
import MyProfile from "./components/core/Dashboard/MyProfile"
import Settings from "./components/core/Dashboard/Settings"
import VideoDetails from "./components/core/ViewCourse/VideoDetails"
```

```
import About from "./pages/About"
import Catalog from "./pages/Catalog"
import Contact from "./pages/Contact"
import CourseDetails from "./pages/CourseDetails"
import Dashboard from "./pages/Dashboard"
import Error from "./pages/Error"
import ForgotPassword from "./pages/ForgotPassword"
// Pages
import Home from "./pages/Home"
import Login from "./pages/Login"
import Signup from "./pages/Signup"
import UpdatePassword from "./pages/UpdatePassword"
import VerifyEmail from "./pages/VerifyEmail"
import ViewCourse from "./pages/ViewCourse"
import { getUserDetails } from "./services/operations/profileAPI"
import { ACCOUNT_TYPE } from "./utils/constants"

function App() {
  const dispatch = useDispatch()
  const navigate = useNavigate()
  const { user } = useSelector((state) => state.profile)
```

```
  useEffect(() => {
    if (localStorage.getItem("token")) {
      const token = JSON.parse(localStorage.getItem("token"))
      dispatch(getUserDetails(token, navigate))
    }
    // eslint-disable-next-line react-hooks/exhaustive-deps
  }, [])

  return (
    <div className="flex min-h-screen w-screen flex-col bg-richblack-900 font-inter">
      <Navbar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/contact" element={<Contact />} />
        <Route path="courses/:courseId" element={<CourseDetails />} />
        <Route path="catalog/:catalogName" element={<Catalog />} />
        {/* Open Route - for Only Non Logged in User */}
        <Route
```

```
  path="login"
  element={
    <OpenRoute>
      <Login />
    </OpenRoute>
  }
/>
<Route
  path="forgot-password"
  element={
    <OpenRoute>
      <ForgotPassword />
    </OpenRoute>
  }
/>
<Route
  path="update-password/:id"
  element={
    <OpenRoute>
      <UpdatePassword />
    </OpenRoute>
```

```
  }
/>
<Route
  path="signup"
  element={
    <OpenRoute>
      <Signup />
    </OpenRoute>
  }
/>
<Route
  path="verify-email"
  element={
    <OpenRoute>
      <VerifyEmail />
    </OpenRoute>
  }
/>
{/* Private Route - for Only Logged in User */}
<Route
  element={
```

```
    <PrivateRoute>
      <Dashboard />
    </PrivateRoute>
  }
>
  {/* Route for all users */}
  <Route path="dashboard/my-profile" element={<MyProfile />} />
  <Route path="dashboard/Settings" element={<Settings />} />
  {/* Route only for Instructors */}
  {user?.accountType === ACCOUNT_TYPE.INSTRUCTOR && (
    <>
      <Route path="dashboard/instructor" element={<Instructor />} />
      <Route path="dashboard/my-courses" element={<MyCourses />} />
      <Route path="dashboard/add-course" element={<AddCourse />} />
      <Route
        path="dashboard/edit-course/:courseId"
        element={<EditCourse />}
      />
    </>
  )}
  {/* Route only for Students */}
```

```
  {user?.accountType === ACCOUNT_TYPE.STUDENT && (
    <>
      <Route
        path="dashboard/enrolled-courses"
        element={<EnrolledCourses />}
      />
      <Route path="/dashboard/cart" element={<Cart />} />
    </>
  )}
  <Route path="dashboard/settings" element={<Settings />} />
</Route>

{/* For the watching course lectures */}
<Route
  element={
    <PrivateRoute>
      <ViewCourse />
    </PrivateRoute>
  }
>
  {user?.accountType === ACCOUNT_TYPE.STUDENT && (
```

```
            <>
              <Route
                path="view-course/:courseId/section/:sectionId/sub-section/:subSectionId"
                element={<VideoDetails />}
              />
            </>
          )}
        </Route>

        {/* 404 Page */}
        <Route path="*" element={<Error />} />
      </Routes>
    </div>
  )
}

export default App
```

CONCLUSION

In conclusion, this document outlines the architecture, features, and functionalities of the StudyNotion ed-tech platform. It highlights the use of MERN stack technologies and REST API design and outlines the deployment process using free hosting services, Vercel for the front-end, Render.com or Railway app for the backend, and MongoDB Atlas for the database. Additionally, it lists potential future enhancements that could be implemented to improve the platform, along with their estimated timelines and priorities.

Throughout the development of the project, various achievements will be made in terms of implementing the desired functionalities and creating a user-friendly interface. However, there will be challenges to be faced during the development process, such as integrating different technologies and debugging errors.

REFERENCE

HTML : https://www.w3schools.com/html/default.asp

CSS : https://www.w3schools.com/css/default.asp

JavaScript : https://developer.mozilla.org/en-US/docs/Web/javascript

React : https://react.dev/

Node JS : https://nodejs.org/en

Express : https://expressjs.com/

MongoDB : https://www.mongodb.com/

Deployment : https://vercel.com/