SYSTEM ENGINEERING PROJECT

Submitted by:
-Anukalp Bhardwaj

TABLE OF CONTENTS

StudyNotion

PROJECT DESCRIPTION

- ❑ Introduction
- ❑ Objectives
- ❑ Features
- ❑ Importance
- ❑ API Creation and Testing

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.

OBJECTIVES

StudyNotion

- 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.

FEATURES

StudyNotion

- For Instructors:
  - Sign Up: Easily register for an account with necessary details.
  - Course Creation: Craft and curate courses with diverse content formats.
  - Course Publication: Effortlessly publish courses for student accessibility.

FEATURES

StudyNotion

- For Students:
    - Registration: Swiftly create accounts with required information.
    - Course Exploration: Access a variety of courses to explore diverse subjects.
    - Course Purchase: Seamlessly select and purchase courses aligned with interests.
    - Content Accessibility: Instant access to course content, including video lectures and reading materials, for an enriching learning experience.

IMPORTANCE

StudyNotion

- StudyNotion is a versatile and intuitive edtech platform that is designed to provide an immersive learning experience to students and a platform for instructors to showcase their expertise.

- StudyNotion's project importance lies in its commitment to continuous improvement. By detailing future enhancements, their impact, and implementation timelines, it ensures innovation aligned with user needs, fostering a dynamic learning environment.

API Creation and Testing

StudyNotion

- The heart of our project lies in the exploration of API creation and testing.
- The StudyNotion platform API is designed following the REST architectural style.
- The sample API requests and responses provided above illustrate how each endpoint will function and what kind of data it will accept or return.
- By leveraging tools like Postman API we demonstrate the seamless functionality of APIs, ensuring they deliver accurate responses and meet the platform’s requirements effectively.

System Design

Architecture Overview

Overview of full stack MERN architecture used in StudyNotion platform.

System Components

Explaining major components and various interactions.

Data Flow

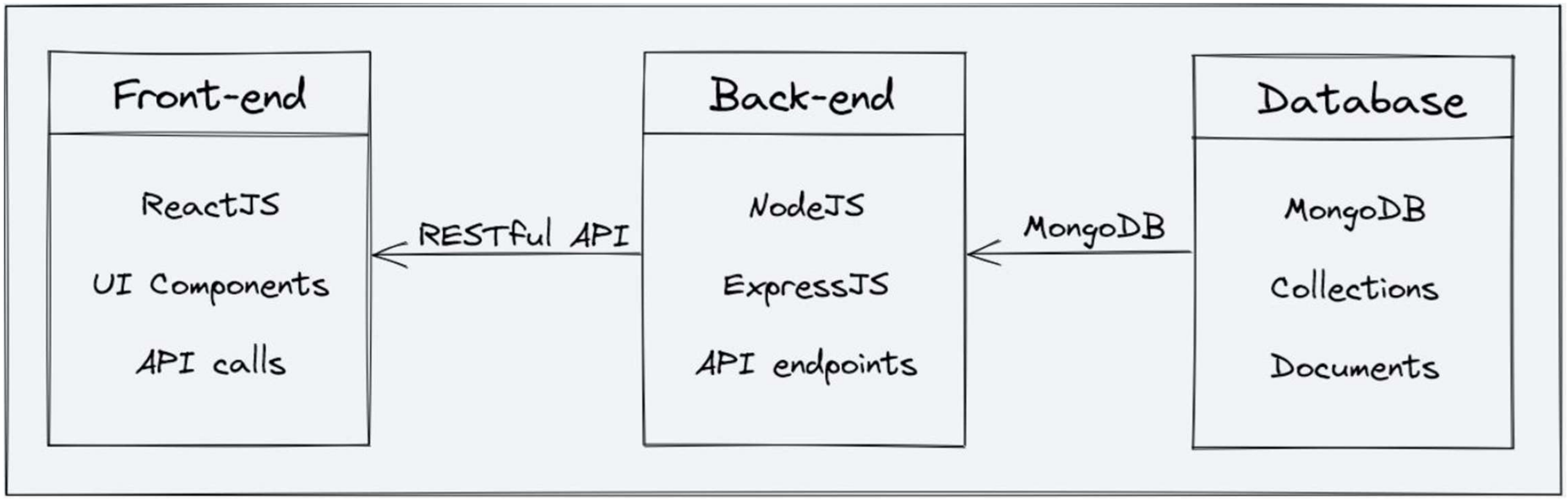
Illustration of flow of data within the system, mainly from user interaction to database storage.

Security Measures

Discussing the security measures implemented in the system to protect user data and ensure secure interactions.

Architecture Overview

Here is a high-level diagram that illustrates the architecture of the StudyNotion ed-tech platform:



System Components

- Backend
- APIs

System Component(Backend)

StudyNotion

Description of Backend Architecture

- Built using Node.js, Express,js and MongoDB as primary database.
- Node.js is a popular JavaScript runtime that allows JS code to run outside 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.

System Component(Backend)

Features and Functionality

- 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.

- Course management: Instructors can create, read, update, and delete courses, as well as manage course content and media. Students can view and rate courses.
- Payment Integration: Students will purchase and enroll on courses by completing the checkout flow that is followed by Stripe integration for payment handling.
- 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.

System Component(Backend)

StudyNotion

Library and Tools used:

- JWT: JSON Web Tokens are used for authentication and authorization, providing a secure and reliable way to manage user credential.
- Bcrypt: It is used for password hashing, adding and extra layer of security to user password.
- Mongoose: It is used as an Object Data Modeling Library, providing a way to interact with MongoDB using JavaScript.
- Cloudinary: It is used to store and manage media related content.
- Dotenv: It is used for storing secrets.
- Nodemailer: It is used to send automatic email after triggering an event.
- Stripe: It is used for Payment gateway process.

System Components(APIs)

The StudyNotion platform API is designed following the REST architectural style. The API is implemented using Node.js and Express.js. It uses JSON for data exchange and follows standard HTTP request methods such as GET, POST, PUT, and DELETE.

API: Application Programming Interface
REST: Representational State Transfer

System Components(APIs)

StudyNotion

Sample list of endpoints and their functionalities:

- /api/auth/signup (POST) - Create a new user (student or instructor) account.
- /api/auth/login (POST) – Log in using existing credentials and generate a JWT token.
- /api/auth/verify-otp (POST) - Verify the OTP sent to the user's registered email.
- /api/auth/forgot-password (POST) - Send an email with a password reset link to the registered email.
- /api/courses (GET) - Get a list of all available courses.
- /api/courses/:id (GET) - Get details of a specific course by ID.
- /api/courses (POST) - Create a new course.
- /api/courses/:id (PUT) - Update an existing course by ID.
- /api/courses/:id (DELETE) - Delete a course by ID.
- /api/courses/:id/rate (POST) - Add a rating (out of 5) to a course.

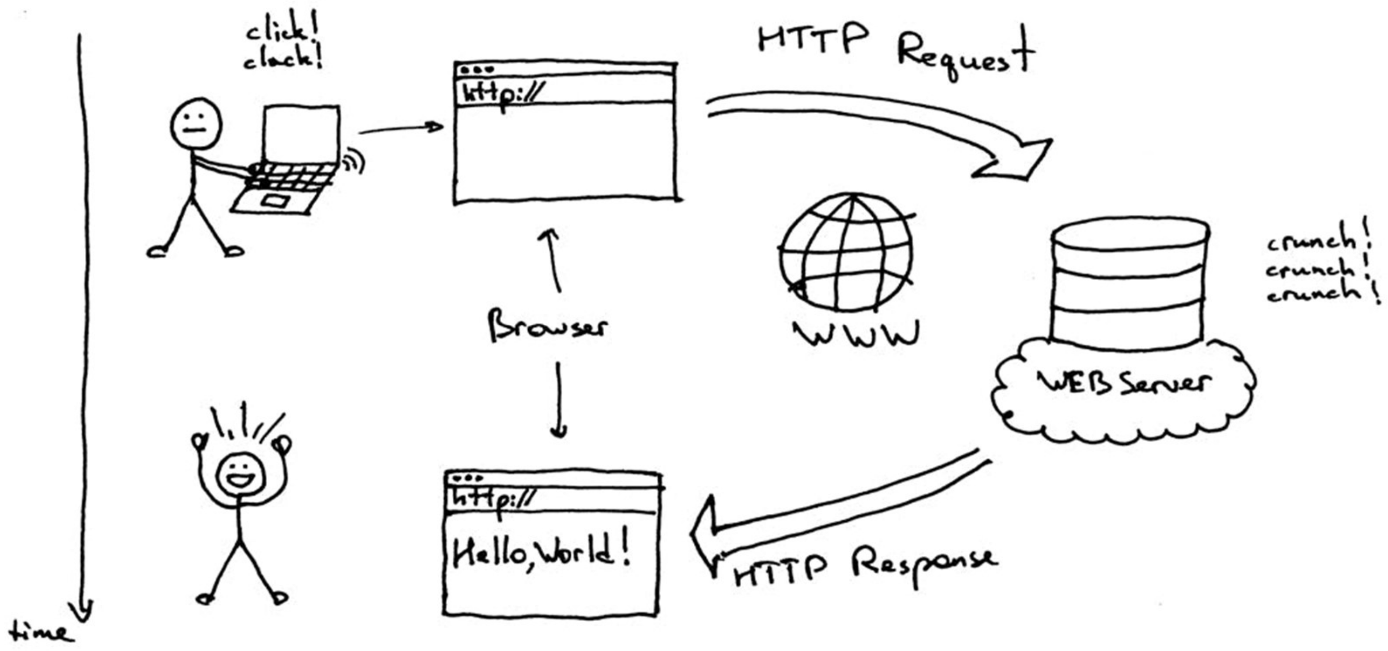
System Components(APIs)

Sample API Requests and Responses:

1. GET /api/courses: Get all courses
   - Response: A list of all courses in the database
2. GET /api/courses/:id: Get a single course by ID
   - Response: The course with the specified ID
3. POST /api/courses: Create a new course
   - Request: The course details in the request body
   - Response: The newly created course
4. PUT /api/courses/:id: Update an existing course by ID
   - Request: The updated course details in the request body
   - Response: The updated course
5. DELETE /api/courses/:id: Delete a course by ID
   - Response: A success message indicating that the course has been deleted.

Data Flow

Client-Server Communication



Data Flow

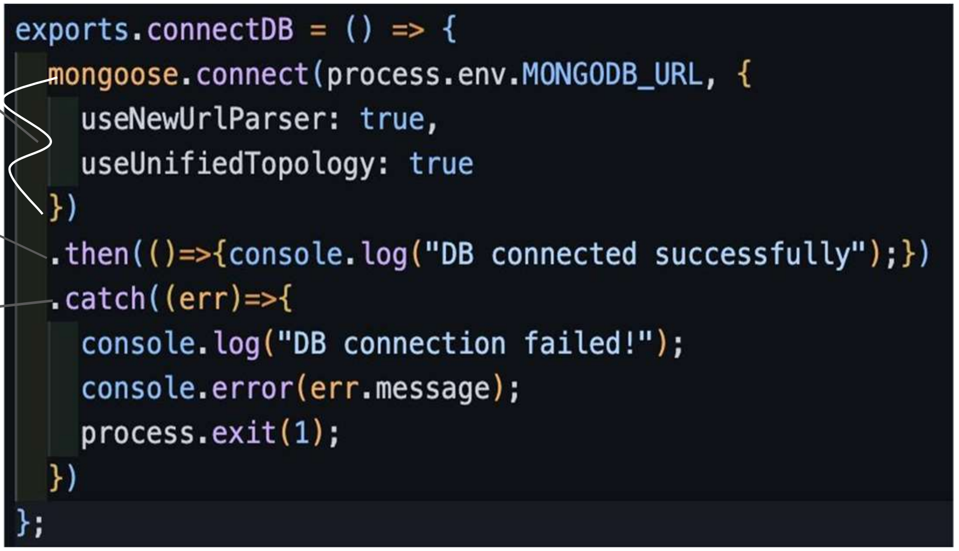
StudyNotion

Database Connectivity

Connect method of mongoose object establishes a connection to the database.

A promise is returned which if resolved then message will be printed

If promise isn't resolved then error message will be printed on the console.



```
exports.connectDB = () => {
  mongoose.connect(process.env.MONGODB_URL, {
    useNewUrlParser: true,
    useUnifiedTopology: true
  })
  .then(()=>{console.log("DB connected successfully");})
  .catch((err)=>{
    console.log("DB connection failed!");
    console.error(err.message);
    process.exit(1);
  })
};
```

MongoDB Compass Demo

Security Measures

StudyNotion

- Authentication
- Authorization
- Data Encryption
- CORS

System Implementation

Technology Stack

Technology used in each layer of system.

Development Process

Development process followed.

Code Structure

Structure of codebase, including folder organization and code patterns.

Code Review

Major code for each part of system.

Technology Stack

- Backend: Node.js and Express.js
- Frontend: React.js
- Database: MongoDB
- Deployment: Frontend(Vercel) & Backend(Render)

Development Process

STAGE 1

Planning and Requirement Analysis

STAGE 2

Defining Requirements

STAGE 3

Design

StudyNotion

Development Process

STAGE 4

Development

StudyNotion

STAGE 5

Testing

STAGE 6

Deployment and Maintenance

Code Structure and Review

Live Demo!

Github Link for Codebase: StudyNotion Codebase
Deployed Website: StudyNotion Website

System Testing

StudyNotion

Approach

End-to-End testing

Test Cases

JSON Objects

Media Files

Tools Used

Postman API

Frontend

Live Testing

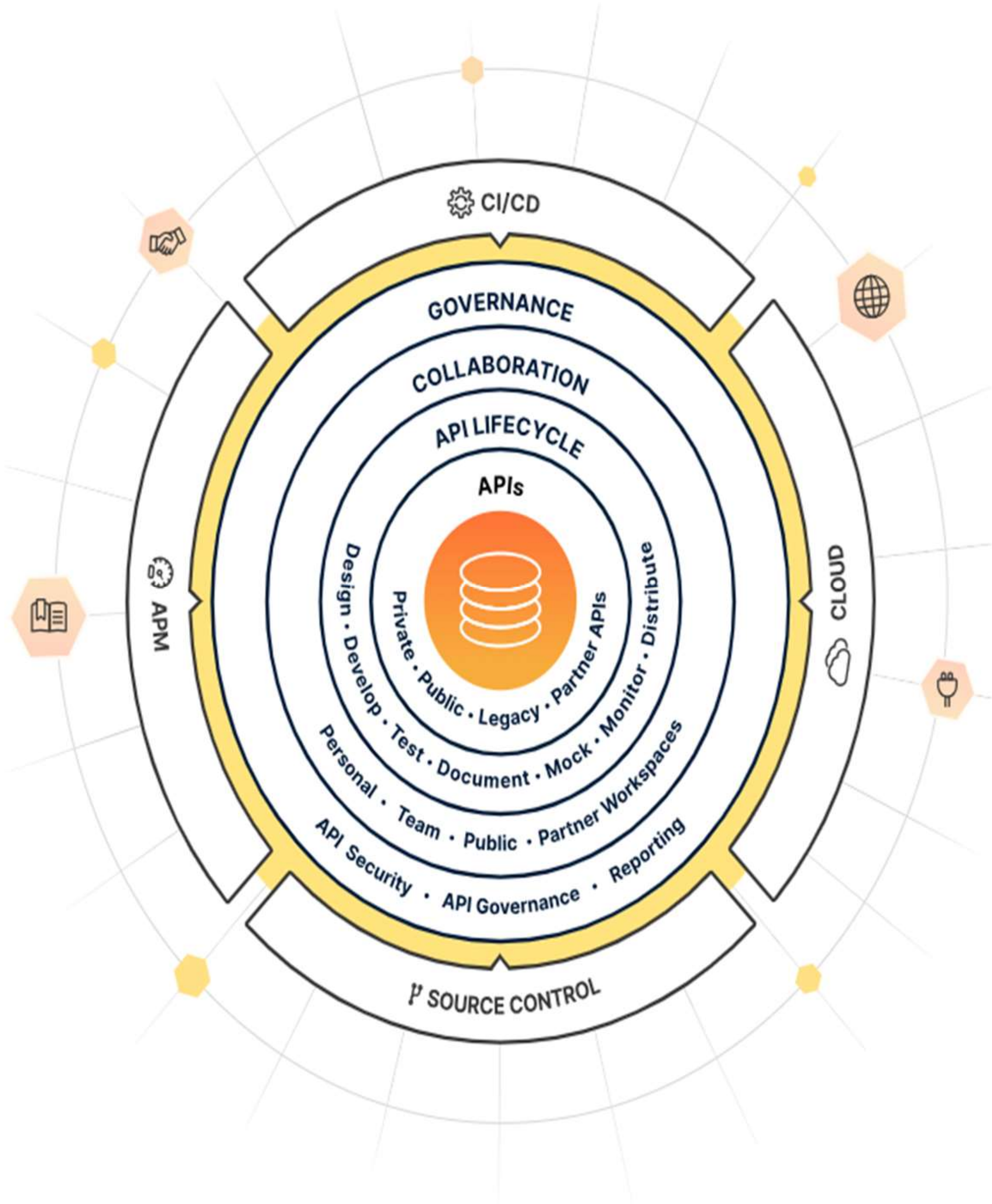
Postman Tutorial

Testing of system

POSTMAN Tutorial

What is Postman?

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.


CI/CD
GOVERNANCE
COLLABORATION
API LIFECYCLE
APIs
Private • Public • Legacy • Partner APIs
Design • Develop • Test • Document • Mock • Monitor • Distribute
Personal • Team • Public • Partner Workspaces
API Security • API Governance • Reporting
APM
CLOUD
SOURCE CONTROL

API REQUESTS COMMON METHODS

- `GET` retrieves data from an API.
- `POST` sends new data to an API.
- `PATCH` and `PUT` update existing data.
- `DELETE` removes existing data.

COLLECTIONS

Collection is a group of API requests

VARIABLES

Elements that can store different values

ENVIRONMENTS

Environment is a set of key value pairs

CONSOLE

console.log()
console.info()

Pre-Request Script

Before a request is sent to the server, as a pre-request script under the Pre-request script tab.

Test Script

After a response is received as a test script under the Tests tab.

System Testing Using Postman API tool

StudyNotion

Live Demo!

Github Link for Codebase: StudyNotion Codebase
Deployed Website: StudyNotion Website

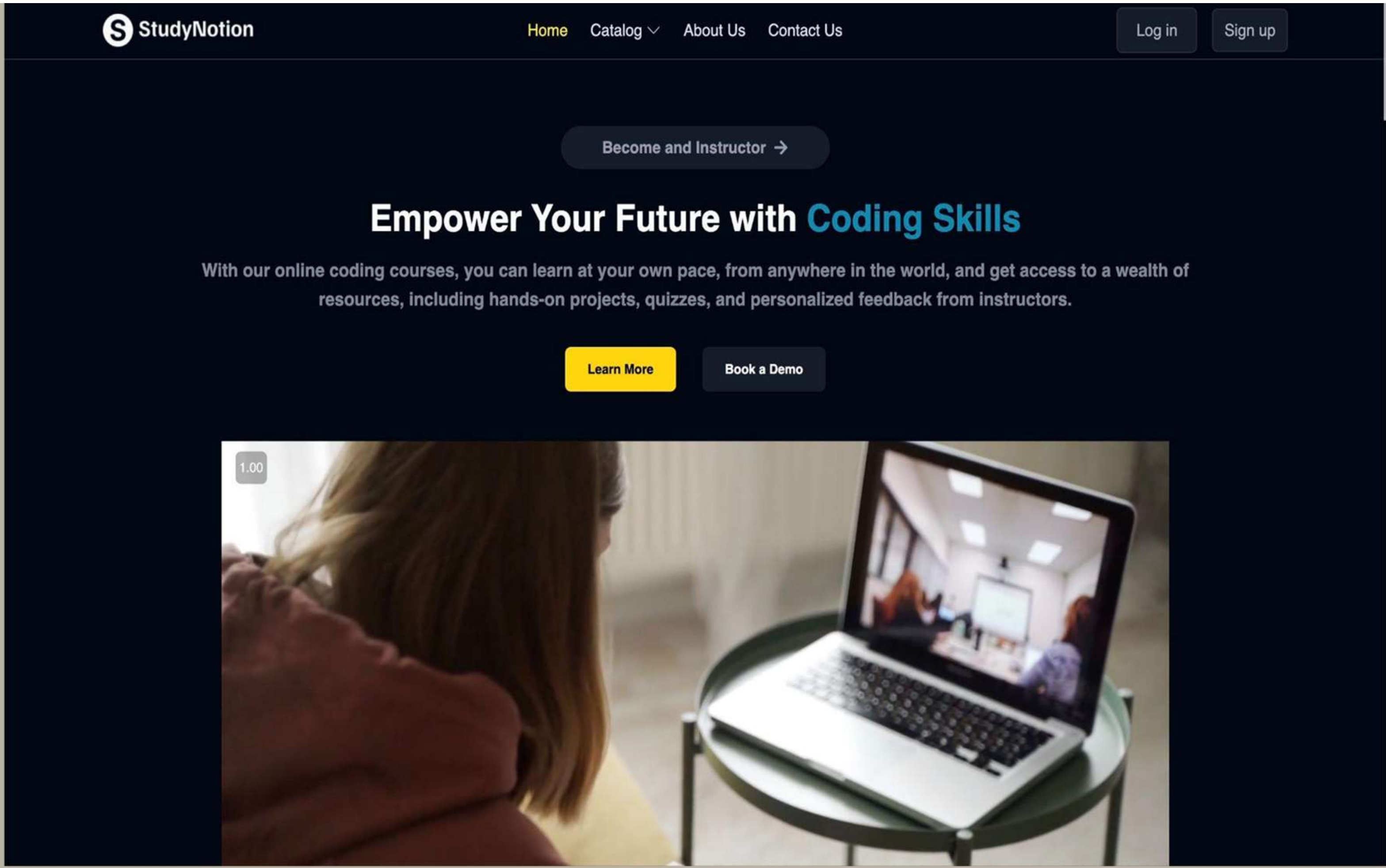
System Testing Using Frontend

Live Demo!

Github Link for Codebase: StudyNotion Codebase
Deployed Website: StudyNotion Website

THANKS

StudyNotion



StudyNotion

Home Catalog About Us Contact Us Log in Sign up

Become and Instructor →

Empower Your Future with Coding Skills

With our online coding courses, you can learn at your own pace, from anywhere in the world, and get access to a wealth of resources, including hands-on projects, quizzes, and personalized feedback from instructors.

Learn More Book a Demo

Github Link for Codebase: StudyNotion Codebase
Deployed Website: StudyNotion Website