EduNexus: A Secure and Scalable MERN-Based Educational Ecosystem with Integrated Payment Authentication

Diya Patil
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology and Management
Mumbai, India
diyapatil226@nhitm.ac.in

Shravani Patane
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology and Management
Mumbai, India
shravanipatane226@nhitm.ac.in

Yash Vadke
Department of Artificial Intelligence and Data Science
New Horizon Institute of Technology and Management
Mumbai, India
yashvadke226@nhitm.ac.in

Abstract— Educational platforms are usually spread across systems for delivering courses, managing hackathons, listing internships and processing payments. These systems often don't have security and strong ways to check transactions. This paper talks about EduN- exus, a scalable educational ecosystem built using the MERN stack, which includes MongoDB, Express.js, React.js and Node.js. EduN- exus brings together student and admin login systems with role-based access control to ensure system management is controlled. Login is done using JSON Web Tokens and RESTful APIs make backend communication structured. Financial transactions are secure through Razorpay integration [16] with HMAC-SHA256, signature verification on the server-side to prevent payment tampering and unauthorized confirmations. The platform allows for course enrollment, registration, browsing internships, managing content and interacting with a chatbot in one unified system. Security considerations include handling credentials role-based authorization protecting API routes and managing secrets based on the environment. Tests show that the system has fast/optimal API response times, reliable payment validation secure multi-role authentication and stable performance under moderate concurrent usage. EduNexus offers an extensible and security-focused foundation, for future educational platforms.

Index Terms— MERN Stack, Educational Platform, Secure Payment Gateway, REST API Architecture, HMAC Verification

I. INTRODUCTION

The education sector is changing fast with schools and universities using digital platforms to deliver courses, manage events and offer internships opportunities. However, many of these platforms work separately so users have to jump between systems to get what they need. Also some platforms don't have security, which makes them easy, targets for scams and identity theft. Traditional systems usually rely on checking payments on the users side which is not enough to stop people from cheating or making unauthorized changes. To fix these problems we need a platform that's unified, secure and can grow with demand. EduNexus is a kind of educational platform that brings together course management, hackathon sign-ups, internship listings and secure payments all in one place. The system has logins for both students and administrators so everyone can manage the system and access it based on their role.

Here are the main things this work achieves:

1. A full-stack educational platform was built using the MERN stack, which includes MongoDB, Express.js, React.js and Node.js.
2. Secure payment verification was implemented on the server-side using HMAC-SHA256 signature validation.
3. JWT-based authentication was integrated to ensure that users and administrators can access the system securely without needing to store sessions.
4. Modular RESTful APIs were designed to support a structured backend.
5. The system was thoroughly tested to evaluate its performance, authentication reliability, transaction security and stability under concurrent usage.

This approach ensures that the platform is secure for roles handles transactions reliably and performs well all within a unified academic ecosystem called EduNexus. The education sector can benefit from a platform like EduNexus. It provides an environment, for students and administrators. With EduNexus users can access educational services in one place.

II. RELATED WORK

When we look at learning we see that platforms such as Coursera [17] and Udemy [18] provide structured course delivery. Then we have websites like Devfolio [19] and HackerEarth [20] that specialize in hackathon and event management. We also have job and internship websites that only focus on listing jobs and internships, even though these websites are good on their own they do not work together to provide a platform that does everything, including managing courses, registering for events, listing internships, controlling administration and handling money transactions securely.

Websites that do lots of things are usually built using MERN stack (MongoDB, Express.js, React.js and Node.js). This set of tools helps make websites that are fast easy to maintain and can handle a lot of users. People who have studied how to build websites say that it is very important to design the system in layers and use APIs to make the website easy to maintain and scalable. However most of the existing research has primarily focused on making the user interface fast and the backend scalable, but not enough on making sure transactions are secure and only allowed people can do administrative tasks in educational websites.

We use things like JSON Web Tokens to make sure users are who they say they are when they are using a website. JWT-based authentication is preferred over session-based approaches as it is stateless and more scalable. We also use something called Role-Based Access Control to make sure that only the right people can do tasks, which helps keep the website safe.

When we pay for things online the website uses codes to make sure the transaction is real and honest. Even though payment websites provide tools to make this work many websites do not

check enough on the server side to make sure everything is okay. There is not a lot of research on how to make sure payments are secure and how to control who can do what in websites.
This is a problem because we need a website that's secure can handle a lot of users and can do everything an educational website needs to do including letting different types of users do different things controlling administration and making sure transactions are secure. This is what the EduNexus platform is trying to do.

1. It will provide a platform, for educational activities
2. It will make sure that only the right people can do tasks
3. It will handle money transactions securely
4. It will be able to handle a lot of users

The EduNexus platform is trying to solve this problem by making an integrated educational ecosystem. Table I presents a feature-wise comparison of EduNexus with existing platforms.

TABLE I. Feature-wise comparison of EduNexus with existing platforms

Feature	**Coursera**	**Udemy**	**Devfolio**	**Edunexus**
Course Management	✓	✓	✗	✓
Hackathon Management	✗	✗	✓	✓
Internship listings	✗	✗	✗	✓
Secure payment gateway	✓	✓	✗	✓
Server-side HMAC verification	✗	✗	✗	✓
Role-based access control	✓	✗	✗	✓
JWT authentication	✓	✗	✗	✓
Unified platform	✗	✗	✗	✓

III. PROPOSED METHODOLOGY

The technologies used in the development of EduNexus are summarized in Table II.

TABLE II. Technology stack used in EduNexus

Technology	**Version**	**Purpose**
React.js	18.x	Frontend UI and component rendering
Node.js	18.x LTS	Backend runtime environment
Express.js	4.x	RESTful API routing and middleware
MongoDB	6.x	NoSQL database for data storage
Razorpay	Latest	Payment gatewayand transaction handling
JSON Web Token	9.x	Stateless authentication and authentication

The EduNexus system is designed as a secure and user-friendly educational platform that integrates multiple academic services within a unified architecture. The primary objective of the system is to ensure that both students and administrators can securely access resources, manage educational content, and perform financial transactions without operational issues.
The platform follows a layered architecture consisting of a frontend interface, a backend server, and a database layer. The frontend, developed using React.js, provides separate login access for students and administrators. Students can enroll in courses, register for hackathons, browse internships, and complete payments, while administrators can securely manage

courses, events, and platform data through controlled access mechanisms.
The backend, implemented using Node.js and Express.js [21], handles all business logic and system operations. It includes modular routes for authentication, course management, event registration, internship handling, payment processing, and administrative control. When a user or administrator logs in, the system generates a JSON Web Token (JWT) [23] that verifies identity. This token is validated for every protected request to ensure that only authorized users can access specific resources. Role-based access control further differentiates permissions between student users and administrators, restricting sensitive operations to authorized admin accounts. The JWT-based authentication and role-based access control flow is shown in Fig. 1.

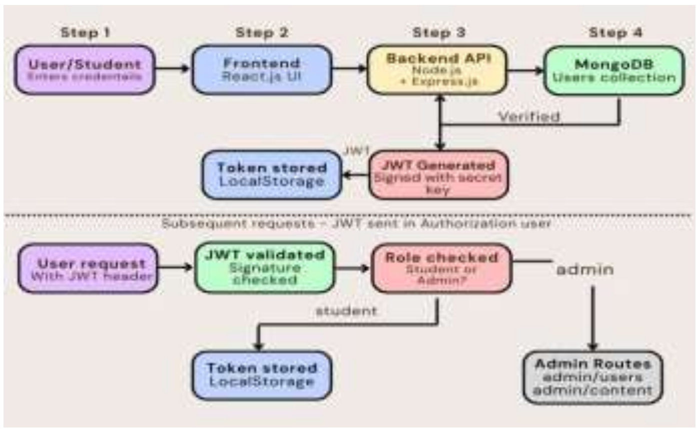


Fig.1 JWT-based authentication and role-based access control flow in EduNexus

Secure payment processing is achieved through integration with Razorpay. When a user initiates a payment, the backend generates a payment order and processes the transaction through the Razorpay interface. After completion, the backend verifies the transaction using HMAC-SHA256 signature validation to ensure authenticity and prevent tampering. The complete payment processing and HMAC-SHA256 signature verification flow is depicted in Fig. 2.

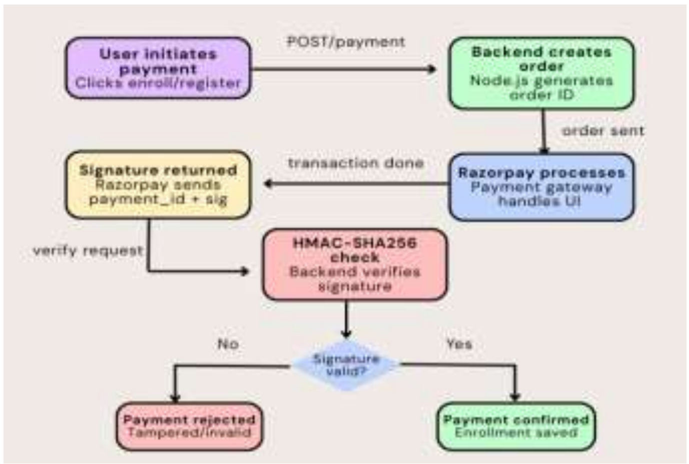


Fig.2. Razorpay payment processing and HMAC-SHA256 signature verification flow in EduNexus

MongoDB is used as the database layer to store system information efficiently. Separate collections are maintained for users, administrators, courses, hackathons, internships, and payment records. The flexible schema design supports scalability and dynamic data management as the platform expands.

The system is built using modular components that work cohesively, allowing easier maintenance and future enhancement. Emphasis has been placed on security, controlled administrative access, and reliable transaction handling. The architecture ensures stable performance under concurrent usage while maintaining secure multi-role authentication and protected data access. Overall, EduNexus provides a secure, scalable, and integrated solution for managing modern educational services.

IV. SYSTEM ARCHITECTURE

The EduNexus platform follows a simple and structured three-layer architecture that helps keep the system secure, scalable, and easy to maintain. These three layers include the Presentation Layer (frontend), the Application Layer (backend), and the Data Layer (database). The system also integrates a payment gateway to handle secure transactions. The platform supports both user and administrative login to ensure controlled system management.

a. Presentation Layer (Frontend – React.js)

The presentation layer is built using React.js. This is where users interact with our system. Both students and administrators can log in to their accounts Students can sign up, look at courses, register for hackathons, check out internships and make payments. Administrators can securely access management features to monitor and control what's on the platform. This layer handles what users see and do. It collects their input, shows content, sends requests to the backend using APIs and manages special tokens like JWT. The frontend talks to the backend through connections and does not directly connect to the database. This makes the whole system more secure.

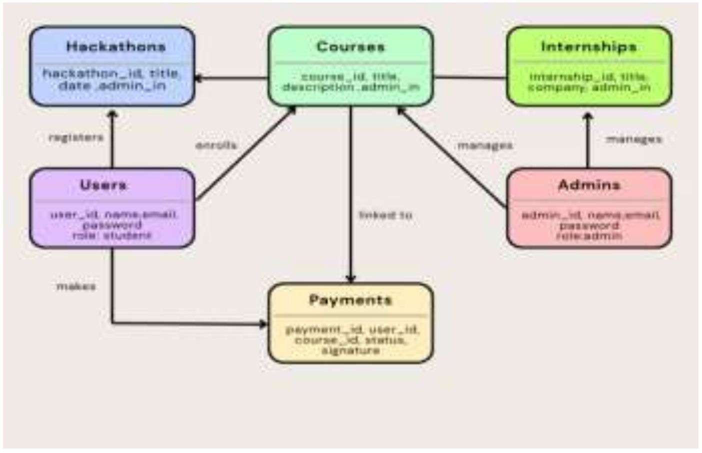
b. Application Layer (Backend – Node.js + Express.js)

The backend is the part of EduNexus. It's built using Node.js and Express.js which handles the business logic and operations of the system. The backend is divided into API routes which includes authentication, course management, hackathon registration, internships, payments, chatbot services and administrative control. To support both users and admins the backend uses role-based authentication. It checks credentials, generates JWT tokens and makes sure authorized admins can access certain operations.
All requests from the frontend are authenticated before any data is stored or retrieved.
The modular API routes implemented in the backend are summarized in Table III.

TABLE III. RESTful API endpoints and access control

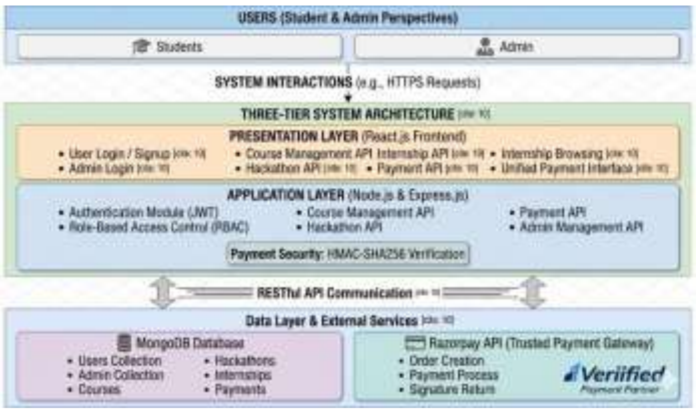
Endpoint	Method	Description	Access
/api/auth/register	POST	Register a new student account	Public
/api/auth/login	POST	Login and receive JWT token	Public
/api/courses	GET	Fetch all available courses	Student
/api/courses/enroll	POST	Enroll student in a course	Student
/api/payment/order	POST	Create Razorpay payment order	Student
/api/payment/verify	POST	Verify HMAC-SHA256 signature	Student
/api/hackathons	GET	Fetch all hackathon listings	Student
/api/admin/courses	POST	Admin adds a new course	Admin
/api/admin/users	GET	Admin views all registered users	Admin

c. Data Layer (MongoDB)
The data layer uses MongoDB to store system information which keeps things organized and scalable. There are collections for users, administrators, courses, hackathons, internships and payment records. MongoDB is used because of its design. It supports data and can scale for the future. The MongoDB collection structure and inter-collection relationships are presented in Fig. 3.


Hackathons
hackathon_id, title, date ,admin_in
Courses
course_id, title, description ,admin_in
Internships
internship_id, title, company, admin_in
registers
enrolls
manages
manages
Users
user_id, name,email, password role: student
linked to
Admins
admin_id, name,email, password role:admin
makes
Payments
payment_id, user_id, course_id, status, signature
Fig 3. MongoDB collection structure and inter-collection relationships in EduNexus

d. External Payment Integration (Razorpay)
EduNexus works with Razorpay for financial transactions. When a user initiates a payment the backend generates a payment order and frontend completes the transaction using Razorpay's interface. After payment, Razorpay returns transaction details and a digital signature, then backend checks this signature using HMAC-SHA256 to confirm it's real. This prevents tampered payment confirmations.

e. Security Design
Security is important throughout the system. It protects both user and administrative operations. JWT-based authentication ensures authorized users and administrators can access protected features. Role-based access control separates student and admin permissions. It ensures administrative actions are restricted to verified admin accounts. Payment signatures are verified on the server to prevent fraud. Sensitive information, like database credentials and payment keys are stored securely using environment variables. The three-layer architecture of EduNexus is illustrated in Fig. 4.


USERS (Student & Admin Perspectives)
Students
Admin
SYSTEM INTERACTIONS (e.g., HTTPS Requests)
THREE-TIER SYSTEM ARCHITECTURE
PRESENTATION LAYER (React.js Frontend)
• User Login / Signup
• Admin Login
• Course Management API Internship API
• Hackathon API
• Payment API
• Internship Browsing
• Unified Payment Interface
APPLICATION LAYER (Node.js & Express.js)
• Authentication Module (JWT)
• Role-Based Access Control (RBAC)
• Course Management API
• Hackathon API
• Payment API
• Admin Management API
Payment Security: HMAC-SHA256 Verification
RESTful API Communication
Data Layer & External Services
MongoDB Database
• Users Collection
• Admin Collection
• Courses
• Hackathons
• Internships
• Payments
Razorpay API (Trusted Payment Gateway)
• Order Creation
• Payment Process
• Signature Return
Veriified
Payment Partner
Fig. 4. System architecture of EduNexus showing the three-layer MERN-based design

V. EXPERIMENTAL SETUP

To check if the EduNexus platform is reliable, secure and performs well we did testing on both student and admin features.

The goal was to make sure it works correctly, has secure authentication, role-based authorization validates payments and performs under normal conditions. The platform was built using the MERN stack, which includes React.js for the frontend, Node.js and Express.js for the backend and MongoDB for storing data. We tested it in a development environment where we used Postman to validate API endpoints. Tested the frontend by interacting with it through a browser, also used Razorpay's test mode to simulate transactions without real money.

We tested the platforms features for both users and admins. For students we checked registration, login, course enrollment, registration, internship browsing and payment processing. For admins we tested login, content management and restricted access to certain endpoints. Tested all modules separately to ensure they handle requests properly and update the database accurately.

We also tested authentication and authorization to ensure that authorized people can access certain features. We tried to log in with credentials and confirmed that unauthorized users were blocked. We tested expired and modified JWT tokens to ensure that protected routes were inaccessible without authentication. We paid attention to admin access control by trying to perform administrative operations using standard user accounts and these attempts were successfully denied.

We tested payment verification using Razorpay's sandbox environment. The backend generated payment orders. We completed simulated transactions using test credentials. After each transaction the backend verified the authenticity of the transaction. We also tried to modify payment response parameters to simulate tampering attempts. The system correctly rejected altered signatures.

We evaluated the platforms performance by measuring API response times for endpoints, including authentication, admin access, course management and payment processing. We executed requests under moderate concurrent load conditions. The average API response time was, within limits and the system maintained stability without crashes or data inconsistencies.

Overall our testing confirmed that the enhanced EduNexus platform supports secure -role access, reliable payment validation and stable performance under realistic usage conditions. The EduNexus platform seems to be reliable, secure and performs well.

VI. SECURITY MODEL

The security system of EduNexus is made to keep user information control who can use the system and make sure money transactions are secure. The system now has two types of security checks one for users and one for administrators to make sure the system is managed properly.

When users log in to EduNexus the system checks their details, gives them a JSON Web Token (JWT). The system uses something called JSON Web Tokens to make this happen.

EduNexus also has a way for administrators to log in. This is so they can manage things like courses and system information. The system has rules about what users and administrators can do so not everyone can do everything. Only administrators who are allowed to, can get administrative sections of the platform. If someone is not allowed they will not be able to get in.

When people pay for something, on EduNexus the system makes sure the payment is real. It does this by using a way of checking the payment so people cannot cheat the system. The backend validates payment details using HMAC-SHA256 signature verification. EduNexus keeps information like secret codes and database details in a safe place, so users cannot get to them.

The system has layers of security to keep everything safe. This means that both users and administrators can trust the system to keep their information protected. The security system of EduNexus is always working to keep user information and system details safe.

VII. RESULTS AND DISCUSSION

The testing of EduNexus was a success. All the main parts of the system worked perfectly when we tested them. Things like logging in, enrolling in courses, signing up for hackathons, looking for internships, talking to the chatbot and making payments all worked without any problems. This means that the different parts of the system like the frontend, backend and database were all talking to each other correctly and reliably.

We also tested the login system to see if it was secure. We tried logging in with information and the system said denied access. We even tried using fake login tokens but the system did not let us in. This shows that the way we are using login tokens called JWT tokens is working well to keep the system safe from people who should not be using it. We also tested to see if the system would let people do things they are not supposed to do. We tried to do some tasks with a regular account but the system stopped us. This means that the system is doing a job of controlling who can do what.

We tested the payment system too. We tried making payments and fake payments and the system only accepted the real ones. This shows that the way we are checking payments on the server using something called HMAC-SHA256 is working well to prevent payments.

When we tested how fast the system was we found that it could handle requests in about 150-200 milliseconds. This is within acceptable thresholds for web applications. When we tested it with a lot of requests at the same time the system did not crash or lose any data. This means that the way we built the system using parts is helping it to work efficiently and reliably.

Overall the testing of EduNexus showed that it is a stable and responsive platform, for education. The way we are using login tokens and checking payments is helping to keep the system safe. The way we built the system is also helping it to work well and making it easy to add features in the future. The performance and security evaluation results of EduNexus are presented in Table IV.

TABLE IV. System performance and security results

Evaluation Parameter	Observation/ Result	Interpretation
Module Functionality	100% modules worked correctly	System is stable and operational
Invalid Login Rejection Rate	100% blocked	Authentication is secure
Modified JWT Token Access	0% access granted	Token validation is effective
Unauthorized Admin Access Attempt	0% success	Role-based control is working
Valid Payment Verification	100% verified	Payment processing is reliable
Tampered Payment Detection	100% rejected	HMAC prevents fraud
Average API Response Time	150–200 ms	System performs efficiently
Concurrent Request Stability	Stable up to 40–50 users (simulated)	Architecture supports load
System Crash Rate	0 during testing	Reliable under normal usage

The EduNexus system worked well for both users and administrators. All parts of the system like user login and course enrollment worked as they should. The system also worked well for registration and payment processing.

The people in charge of the system can use the Admin Login to manage what is on the platform. They can do this without letting just anyone get in. The system was tested to make sure that only the right people can get in. If someone tries to get in without the login they will not be able to. The system will stop them from doing anything that only administrators can do.

The system uses codes to keep people from getting in who should not be there. It checks these codes to make sure they are real. If someone tries to get in with a code the system will stop them. The system also checks to make sure that payments are real. It does this by using a kind of check.
The system was tested to see how well it works when a lot of people are using it at the time. It worked well. Did not crash. The system also responded quickly to what people were doing. Administrators of the system can be sure that EduNexus is an reliable system. It lets them control what is on the platform without slowing down the system or making it less safe.

VIII. CONCLUSION AND FUTURE WORK

This paper presented EduNexus, a secure and scalable full-stack educational platform designed using a three-layer MERN-based architecture. The system integrates user and administrative login, role-based access control, modular API design, and secure payment processing within a unified framework.
The system uses JWT-based authentication that verify users. It is also a stateless way means that the system does not have to keep track of user information. Role-based authorization is also used which separates student and administrative privileges. The system also uses server-side HMAC-SHA256 signature validation which makes transactions more secure. It prevents people from tampering with payments. It also prevents unauthorized payment confirmations. The people who made EduNexus tested it. They found out that it works well. It is stable and efficient. The API response times are fast, access control is secure and transaction validation is reliable. All of this is true when a lot of people are using the system. The architecture of EduNexus is modular and layered which makes it easy to maintain. It also makes it easy to add features. Overall EduNexus is an example of a secure educational ecosystem also shows that we can make a multi-role educational platform using modern technologies. We can do this while still maintaining security standards. EduNexus is a platform, for education.
Future work may include the integration of cloud deployment using services such as AWS or Azure to support large-scale user access. Implementing a microservices-based architecture can improve scalability and fault isolation. Advanced monitoring tools can be added for real-time system analytics and performance tracking.
Additionally, implementing advanced security measures such as multi-factor authentication (MFA) and intrusion detection mechanisms can further strengthen system security.
These enhancements will allow EduNexus to evolve into a more intelligent, scalable, and enterprise-ready academic ecosystem.

REFERENCES

[1] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, May 2015. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7519

[2] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC 2104, Internet Engineering Task Force, Feb. 1997. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2104

[3] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Internet Engineering Task Force, Aug. 2008. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5246

[4] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)," FIPS PUB 186-4, Gaithersburg, MD, USA: NIST, Jul. 2013.

[5] National Institute of Standards and Technology (NIST), "Security and Privacy Controls for Information Systems and Organizations," NIST SP 800-53 Rev. 5, Gaithersburg, MD, USA: NIST, Sep. 2020.

[6] A. Barth, C. Jackson, and J. C. Mitchell, "Robust defenses for cross-site request forgery," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Alexandria, VA, USA, 2008, pp. 75–88.

[7] G. DeCandia et al., "Dynamo: Amazon's highly available key-value store," in *Proc. ACM Symp. Operating Systems Principles (SOSP)*, Stevenson, WA, USA, 2007, pp. 205–220.

[8] E. Brewer, "CAP twelve years later: How the 'rules' have changed," *IEEE Computer*, vol. 45, no. 2, pp. 23–29, Feb. 2012.

[9] S. Romanosky, "Examining the costs and causes of cyber incidents," *Journal of Cybersecurity*, vol. 2, no. 2, pp. 121–135, 2016.

[10] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[11] R. T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," Ph.D. dissertation, Dept. Inf. and Comput. Sci., Univ. of California, Irvine, CA, USA, 2000.

[12] L. Richardson and S. Ruby, *RESTful Web Services*. Sebastopol, CA, USA: O'Reilly Media, 2007.

[13] M. Howard and D. LeBlanc, *Writing Secure Code*, 2nd ed. Redmond, WA, USA: Microsoft Press, 2003.

[14] D. Ferraiolo, R. Kuhn, and R. Chandramouli, *Role-Based Access Control*, 2nd ed. Norwood, MA, USA: Artech House, 2007.

[15] MongoDB Inc., "MongoDB Architecture Guide," 2024. [Online]. Available: https://www.mongodb.com/resources/products/fundamentals/mongodb-architecture

[16] Razorpay, "Razorpay API Documentation — Payment Gateway Integration," 2024. [Online]. Available: https://razorpay.com/docs/

[17] Coursera Inc., "Coursera Platform — Online Learning and Courses," 2024. [Online]. Available: https://www.coursera.org

[18] Udemy Inc., "Udemy — Online Courses and Learning Platform," 2024. [Online]. Available: https://www.udemy.com

[19] Devfolio, "Devfolio — Hackathon Management Platform," 2024. [Online]. Available: https://devfolio.co

[20] HackerEarth Inc., "HackerEarth — Developer Assessment and Hackathon Platform," 2024. [Online]. Available: https://www.hackerearth.com

[21] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to build high-performance network programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, Nov.–Dec. 2010.

[22] I. Grigorik, "HTTPS everywhere," in *High Performance Browser Networking*. Sebastopol, CA, USA: O'Reilly Media, 2013, ch. 4.

[23] A. Joshi, S. Kiran, and P. Desai, "Security analysis of JWT-based authentication in web applications," in *Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, 2018, pp. 1–6.