

An  
INTERNSHIP REPORT  
ON  
**Unit Test Automation System**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING  
INFORMATION TECHNOLOGY

**BY**

Nachiket Jadhav  
Roll No: 33134  
Exam Seat No:T190058597

Under the guidance of  
Ms. Deepali Salapurkar



DEPARTMENT OF INFORMATION TECHNOLOGY  
PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
SR. NO 27, PUNE-SATARA ROAD, DHANKAWADI  
PUNE - 411 043.  
AY: 2023-2024

SCTR's PUNE INSTITUTE OF COMPUTER TECHNOLOGY  
DEPARTMENT OF INFORMATION TECHNOLOGY



**C E R T I F I C A T E**

This is to certify that the SPPU Curriculum-based internship report entitled  
“Unit Test Automation System”

Submitted by

Name : Nachiket Jadhav

Roll No: 33134 .

has satisfactorily completed the curriculum-based internship under the guidance  
of Ms. Deepali Salapurkar towards the partial fulfillment of third year Informa-  
tion Technology Semester VI, Academic Year 2023-24 of Savitribai Phule Pune  
University.

Ms. Deepali Salapurkar  
Internship Guide

Dr. A. S. Ghotkar  
HOD IT

Date:  
Place: Pune

## **Acknowledgement**

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. Special thanks is due to my Guide Ms. Deepali Salapurkar and Reviewer Ms.Rachna Karnavat whose help, stimulating suggestions and encouragement helped me in all time of fabrication process and in writing this report. I also sincerely thanks for the time spent proof- reading and correcting my many mistakes I would also like to thank my Internship Mentor Mr. Naman Buradkar to continuously help me through the course of making this report and also providing resources to complete the tasks. Many thanks to the PICT faculty and Dr. A.S Ghotkar Mam head of information technology department for providing me the opportunity to complete my Internship and present this report by providing all the resources required.

Name : Nachiket Jadhav

Roll No: 33134

# Contents

Certificate . . . . .	i
Acknowledgement . . . . .	ii
Contents . . . . .	iii
List of Figures . . . . .	iv
Abbreviations . . . . .	v
<b>1 Introduction</b>	<b>vi</b>
1.1 Introduction . . . . .	vi
<b>2 Problem Statement</b>	<b>1</b>
2.1 Problem Statement: . . . . .	1
<b>3 Objective and Scope</b>	<b>2</b>
3.1 Objectives . . . . .	2
3.2 Scope . . . . .	2
3.3 Motivation . . . . .	3
<b>4 Methodologies</b>	<b>5</b>
4.1 Web Technologies used . . . . .	5
4.2 Frontend Technologies . . . . .	5
4.3 Backend Technologies . . . . .	6
4.4 Integration . . . . .	7
<b>5 Implementation Tools</b>	<b>8</b>
<b>6 Outcome</b>	<b>9</b>
6.1 Frontend . . . . .	9
6.2 Backend . . . . .	11
<b>7 Conclusion</b>	<b>13</b>
<b>References</b>	<b>14</b>
<b>Plagiarism Report</b>	<b>15</b>

# List of Figures

6.1	Admin Login Page . . . . .	9
6.2	Admin Profile . . . . .	9
6.3	Add Student Page . . . . .	10
6.4	Add Faculty Page . . . . .	10
6.5	Student Profile Page . . . . .	11
6.6	Admin Login Details . . . . .	11
6.7	Postman API hit . . . . .	12
6.8	Credentials Schema . . . . .	12

## **Abbreviations**

MERN : MongoDB, ExpressJS, ReactJS, NodeJS

NPM : Node Packet Manager

# **1. Introduction**

## **1.1 Introduction**

In response to the evolving landscape of education, the need for streamlined processes and enhanced accessibility has become increasingly paramount. Educational institutions worldwide are facing a myriad of challenges in traditional examination administration, ranging from cumbersome content management to disparate communication channels and limited student engagement. Within this context, the development of an exam automation website emerges as a pivotal solution aimed at revolutionizing the examination ecosystem and addressing these multifaceted challenges head-on. This report encapsulates the transformative journey undertaken during an internship focused on the conception, design, and implementation of such a platform, tailored specifically for our college community. At the heart of this internship project lies a profound commitment to redefining the educational experience for all stakeholders—teachers, students, and administrators alike. Recognizing the inefficiencies and complexities inherent in traditional examination processes, our primary objective was to leverage the power of modern technology to empower and streamline exam-related activities. Through collaboration, innovation, and meticulous attention to user-centric design principles, we embarked on a journey to create an intuitive and efficient platform that centralizes examination management, communication, and engagement.

## **2. Problem Statement**

### **2.1 Problem Statement:**

Traditional exam management processes are inefficient and error-prone, lacking scalability and security. Educational institutions struggle with exam creation, candidate enrollment, and result processing. There's a need for a comprehensive Exam Management System (EMS) website to streamline these processes, enhance exam integrity, and provide insightful analytics.



## **3. Objective and Scope**

### **3.1 Objectives**

To have a centralized and fully functioning UT exam admin platform that brings ease of communication within the department by getting multiple features onto a single platform. It involves:

- Marks uploading
- Marks viewing
- Content uploading
- Seating arrangement Viewing
- Notice communication with administrators.
- Decoder Loss Weight

### **3.2 Scope**

The scope of this project encompasses the development of a comprehensive exam automation website tailored specifically for our college community. This platform aims to streamline various aspects of the examination process, including content management, result dissemination, practice resources, and communication channels. Key features within the scope of this project include:

- Teacher Functionality:
  - Ability to upload exam content, including questions, answers, and marking schemes.
  - Facility to input and manage marks for students' examinations.
  - Access to administrative tools for scheduling exams and generating reports.
- Student Functionality:
  - Access to view exam results securely and efficiently.
  - Availability of practice questions and resources to aid in exam preparation.

- Integration of notices and announcements relevant to their academic journey.
- Administrator Functionality:
  - Capability to manage user accounts and permissions across different user roles.
  - Ability to create and publish notices for both teachers and students.
  - Oversight of system-wide settings and configurations to ensure optimal performance and usability.
- User Profiles:
  - Creation of distinct user profiles for teachers, students, and administrators.
  - Inclusion of personalized information and preferences to enhance user experience and system functionality.

### 3.3 Motivation

Traditional examination administration often entails laborious and time-consuming processes for teachers, administrators, and students alike. Teachers grapple with the challenge of managing vast amounts of examination content, ranging from questions and answers to marking schemes and assessments. Meanwhile, administrators navigate disparate communication channels to disseminate vital information and notices to teachers and students. Amidst these challenges, students often face limited access to practice resources and struggle to engage meaningfully with their examination preparation. The need for a comprehensive solution to these challenges became increasingly apparent, prompting the inception of the exam automation website project. Our vision was not merely to digitize existing processes but to fundamentally transform the way examinations are administered and experienced within our college community. By harnessing the transformative potential of technology, we sought to catalyze a culture of efficiency, transparency, and empowerment that transcends traditional boundaries. The journey from conceptualization to realization of the exam automation website was characterized by collaboration, innovation, and relentless dedication to our mission. Technical architecture was meticulously designed to ensure scalability, reliability, and performance, leveraging a modern stack of programming languages, frameworks, and databases. User functionalities were carefully crafted to meet the diverse needs and preferences of teachers, students, and administrators, fostering seamless interaction and engagement.

Central to our approach was a commitment to user-centric design principles, ensuring that the exam automation website prioritized usability, accessibility, and intuitiveness. Through iterative prototyping, user feedback sessions, and usability testing, we iteratively refined the user interface to create an engaging and immersive experience for all stakeholders. The result is a visually appealing and functionally robust platform that empowers users to navigate examination processes with ease and confidence. Testing and quality assurance played a pivotal role in ensuring the reliability and robustness of the exam automation website. Rigorous testing methodologies, including unit testing, integration testing, and user acceptance testing, were employed to identify and rectify any potential issues or vulnerabilities. Security and privacy considerations were paramount, with robust measures implemented to safeguard sensitive information and protect user data integrity.

## 4. Methodologies

### 4.1 Web Technologies used

Web technologies used: MERN Stack (MongoDB, Express.js, React.js, Node.js)

The UT automation project leverages the MERN (MongoDB, Express.js, React.js, Node.js) stack, which facilitates versatility and scalability in web application development.

MongoDB, a NoSQL database, offers flexibility in data storage and retrieval, ideal for managing various schemas involved in the process.

Express.js simplifies the creation of robust and efficient server-side applications, providing a framework for building RESTful APIs to handle data communication between the client and server.

React.js, a JavaScript library for building user interfaces, enables the creation of dynamic and interactive UI components, enhancing the user experience for both students and faculty.

Node.js is a runtime environment built on Chrome's V8 JavaScript engine, which allows developers to run JavaScript code outside the web browser. It's known for its non-blocking, event-driven architecture, making it highly efficient and suitable for handling I/O operations, such as file system access and network requests. In the context of the UT automation project it serves as the runtime environment for executing JavaScript code on the server-side, facilitating seamless integration with the frontend components.

Together, the MERN stack empowers the UT automation project with a comprehensive toolkit for developing a sophisticated, user-friendly web application.

### 4.2 Frontend Technologies

Frontend development plays a crucial role in shaping the user experience and interface design. React.js, a key component of the MERN stack, is particularly well-suited for this purpose due to its versatility, performance, and extensive ecosystem of libraries and components.

React.js excels in building dynamic and responsive user interfaces, making it an ideal choice for creating the frontend of the UT automation system. Its component-based architecture allows for the creation of reusable UI elements, which promotes consistency across different parts of the application and simplifies maintenance. This modularity also facilitates collaboration

among frontend developers, enabling them to work on distinct components independently and integrate them seamlessly into the overall system.

Using React.js for the frontend development of the UT automation project offers numerous advantages, including enhanced user experience, improved performance, efficient state management, and compatibility with modern development practices.

### 4.3 Backend Technologies

In the UT automation project utilizing the MERN stack, the backend technology, Node.js with Express.js, plays a pivotal role in handling server-side operations, managing data, and facilitating communication between the frontend and the database.

**Node.js:** Node.js provides the foundation for building scalable and high-performance server-side applications. Node.js's asynchronous, non-blocking nature makes it well-suited for handling concurrent operations, such as database queries and network requests. This improves application performance and scalability, particularly in scenarios where multiple users interact with the system simultaneously. Node.js has a vast ecosystem of packages and modules available through the Node Package Manager (NPM). This allows developers to leverage existing libraries and tools to expedite development, enhance functionality, and address common challenges.

**Express.js:** Express.js is a minimalist web application framework for Node.js, providing a robust set of features for building web servers and APIs. It simplifies the process of handling HTTP requests, routing, middleware integration, and error handling, allowing developers to focus on building core application logic. Express.js is widely used in conjunction with Node.js due to its flexibility, simplicity, and extensive community support.

**MongoDB:** MongoDB is a NoSQL database that offers flexibility, scalability, and performance for handling large volumes of data. It stores data in flexible, JSON-like documents, making it suitable for storing semi-structured and unstructured data. MongoDB's document-based model allows for dynamic schemas, enabling developers to iterate quickly and adapt to changing data requirements. It serves as the backend database for storing student details, faculty information, admin information, and other relevant data.

In conclusion, using Node.js with Express.js and MongoDB as database for the backend of the UT automation project offers numerous benefits, including streamlined development, asynchronous programming capabilities, scalability, and extensive community support. To-

gether, they provide a robust foundation for building high-performance, scalable, and efficient server-side applications to support the needs of the UT automation system.

## 4.4 Integration

**Backend Development with Node.js and Express.js:** Developed the backend logic using Node.js and Express.js to handle API routes, business logic, and database operations. Define endpoints for exam-related functionalities such as scheduling exams, managing student details, and generating reports. Utilized middleware for handling authentication, error handling, and other cross-cutting concerns.

**Database Setup with MongoDB Atlas:** Created a new cluster in MongoDB Atlas and configured database access, network settings, and security options. Obtained the connection string for your MongoDB Atlas cluster(uri) to connect the Node.js backend to the database.

**Integration of MongoDB with Mongoose:** Define Mongoose schemas and models for exam-related data entities such as students, faculty, and admin details information. Established the connection to the MongoDB database using `Mongoose.connect()` method, providing the connection string for MongoDB Atlas.

**State Management with Redux:** Implemented Redux for centralized state management in your React.js application, especially if the application's state is complex or needs to be shared across multiple components. Defined actions and reducers to manage state changes related to student data, and other application features.

**AWS S3:** Amazon Simple Storage Service (S3) is a cloud storage service provided by Amazon Web Services (AWS) that allows users to store and retrieve large amounts of data. In the context of the UT automation project, AWS S3 is utilized for the storage of documents uploaded by administrators or faculty members.

**Frontend Development with React.js:** Developed the frontend user interface using React.js to create interactive and dynamic views for students, faculty, and administrators. Implemented components for displaying exam schedules, exam results for students. Profiles for respective users (student, admin, faculty). Utilized React Router for client-side routing to navigate between different pages and components within the application.

## **5. Implementation Tools**

- Frontend Tools
  - React
  - Redux
  - HTML and CSS
- Backend Tools
  - NodeJS
  - ExpressJS
  - Multer
- Database
  - MongoDB
  - Atlas Mongo
- API checking
  - Postman

## 6. Outcome

### 6.1 Frontend

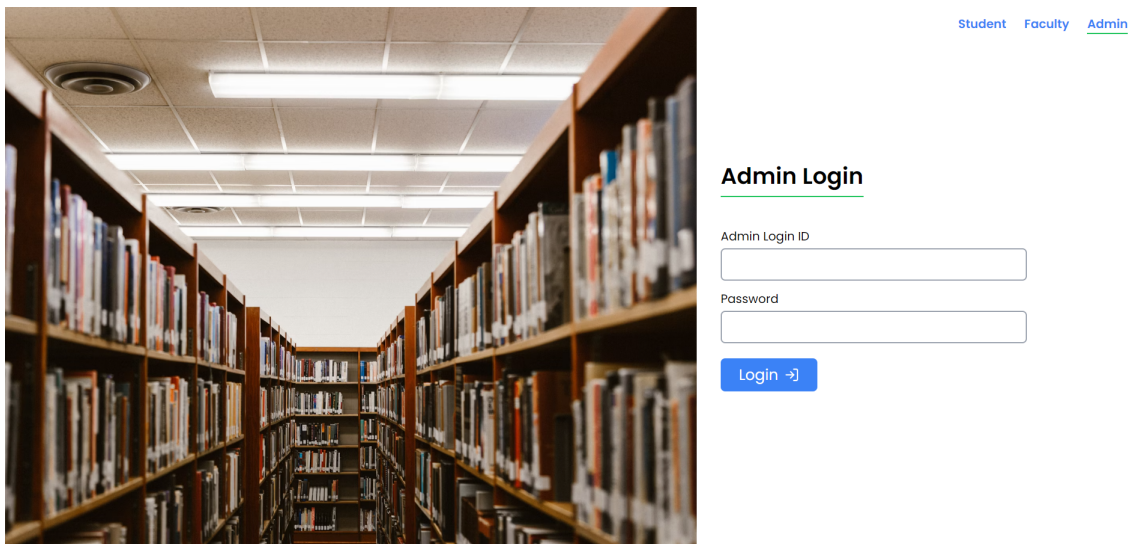


Figure 6.1: Admin Login Page

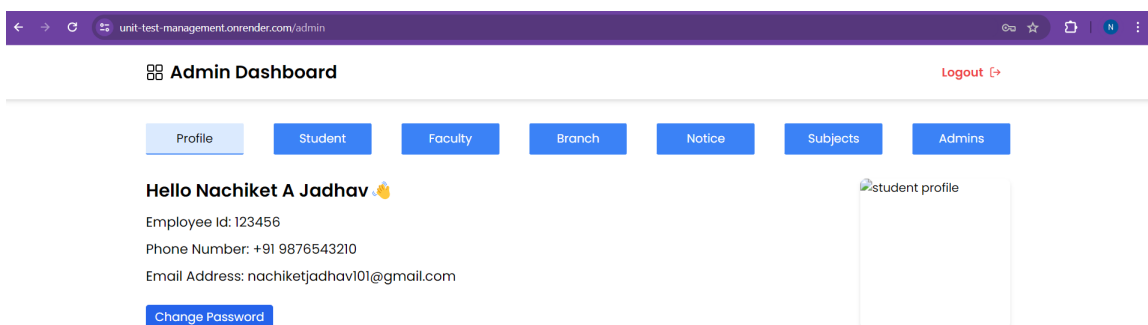


Figure 6.2: Admin Profile



The screenshot shows a web browser window with the URL `unit-test-management.onrender.com/admin`. The navigation bar at the top contains buttons for Profile, Student, Faculty, Branch, Notice, Subjects, and Admins. The main heading is "Student Details". To the right of the heading are links for "Add Student" and "Edit Student". The form contains the following fields:

- Enter First Name (text input)
- Enter Middle Name (text input)
- Enter Last Name (text input)
- Enter Enrollment No (text input)
- Enter Email Address (text input)
- Enter Phone Number (text input)
- Select Semester (dropdown menu, currently showing "-- Select --")
- Select Branch (dropdown menu, currently showing "-- Select --")
- Select Gender (dropdown menu, currently showing "-- Select --")
- Select Profile (text input with an "Upload" button and a file icon)

At the bottom of the form is a blue button labeled "Add New Student".

Figure 6.3: Add Student Page

The screenshot shows a web browser window with the URL `unit-test-management.onrender.com/admin`. The navigation bar at the top contains buttons for Profile, Student, Faculty, Branch, Notice, Subjects, and Admins. The main heading is "Faculty Details". To the right of the heading are links for "Add Faculty" and "Edit Faculty". The form contains the following fields:

- Enter First Name (text input)
- Enter Middle Name (text input)
- Enter Last Name (text input)
- Enter Employee Id (text input)
- Enter Email Address (text input)
- Enter Phone Number (text input)
- Select Department (dropdown menu, currently showing "-- Select --")
- Enter POST (text input)
- Select Gender (dropdown menu, currently showing "-- Select --")
- Enter Experience (text input)
- Select Profile (text input with an "Upload" button and a file icon)

At the bottom of the form is a blue button labeled "Add New Faculty".

Figure 6.4: Add Faculty Page

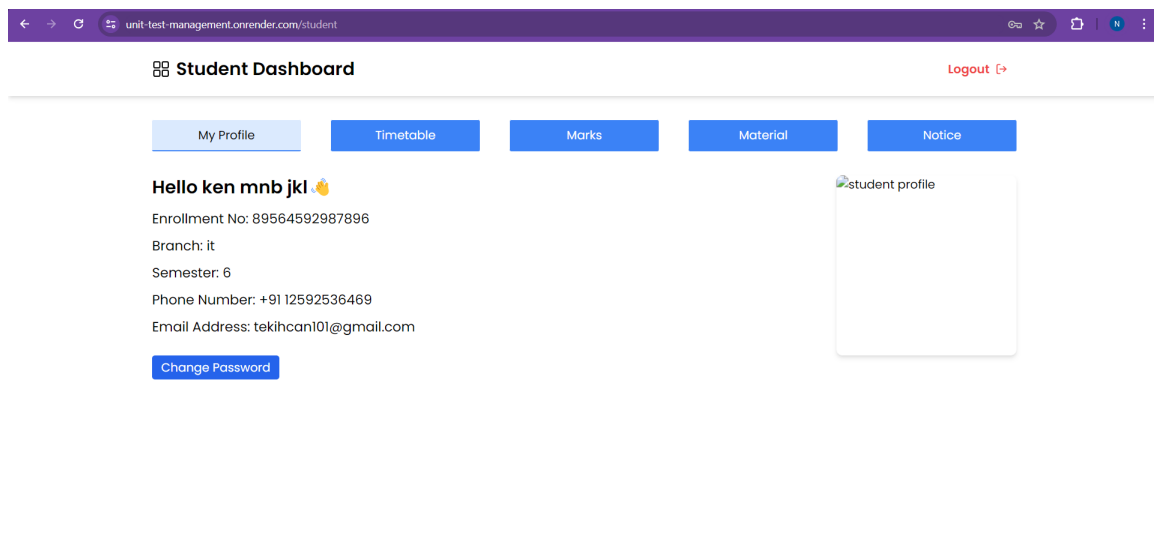


Figure 6.5: Student Profile Page

## 6.2 Backend

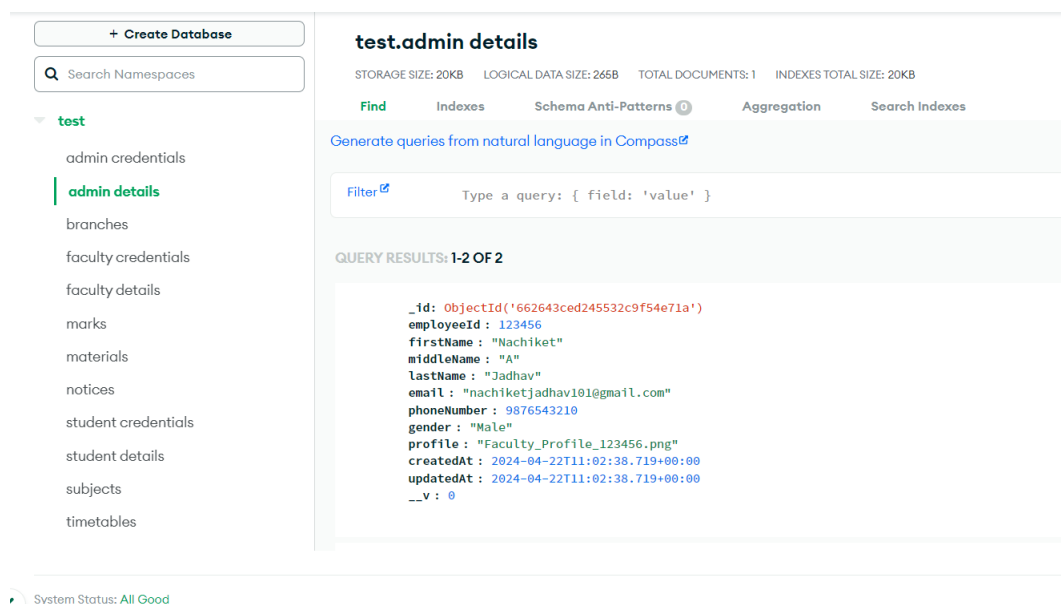


Figure 6.6: Admin Login Details

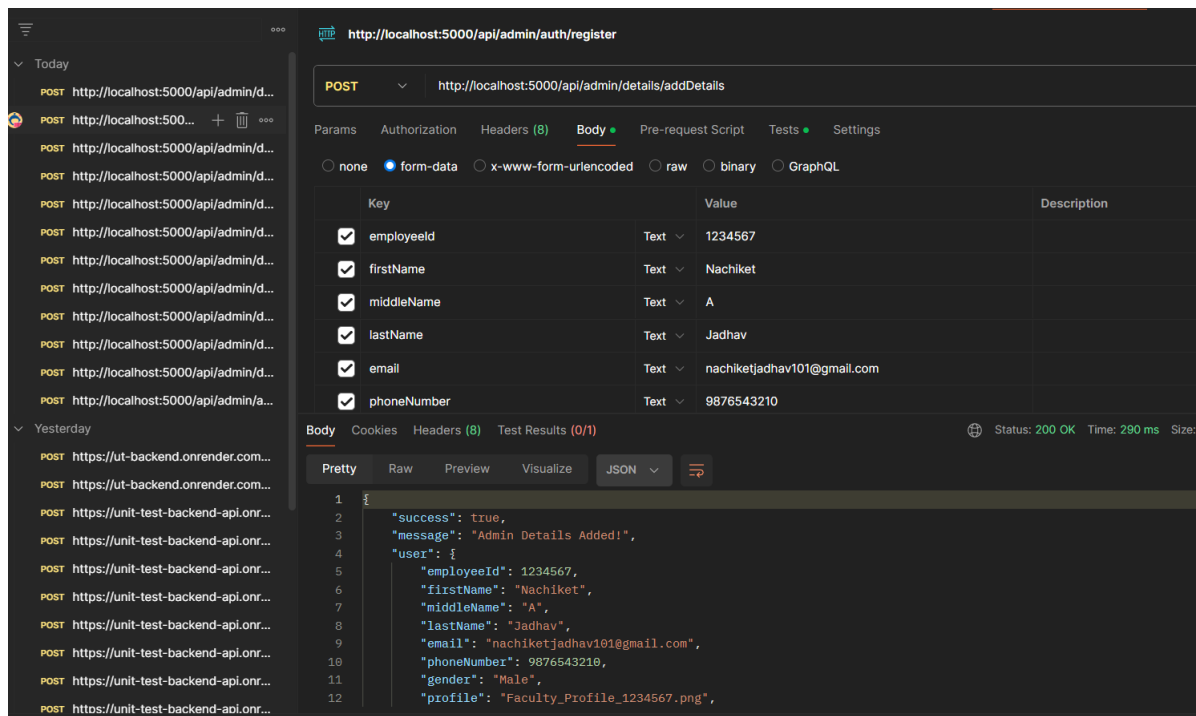


Figure 6.7: Postman API hit

+ Create Database

Q

Search Namespaces

test

- admin credentials
- admin details
- branches
- faculty credentials
- faculty details
- marks
- materials
- notices
- student credentials
- student details
- subjects
- timetables

test

LOGICAL DATA SIZE: 1.1KB   STORAGE SIZE: 144KB   INDEX SIZE: 144KB   TOTAL COLLECTIONS: 12

CREATE COLLECTION

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
admin credentials	1	103B	103B	20KB	1	20KB	20KB
admin details	2	531B	266B	36KB	1	36KB	36KB
branches	1	82B	82B	20KB	1	20KB	20KB
faculty credentials	0	0B	0B	4KB	1	4KB	4KB
faculty details	0	0B	0B	4KB	1	4KB	4KB
marks	0	0B	0B	4KB	1	4KB	4KB
materials	0	0B	0B	4KB	1	4KB	4KB
notices	0	0B	0B	4KB	1	4KB	4KB
student credentials	1	115B	115B	20KB	1	20KB	20KB
student details	1	308B	308B	20KB	1	20KB	20KB

Figure 6.8: Credentials Schema

## **7. Conclusion**

In conclusion, the development of the exam automation website marks a significant milestone in our ongoing efforts to modernize and optimize the examination process within our college community. Through the collaborative endeavours of the internship project, we have succeeded in conceptualizing, designing, and implementing a robust platform that addresses the multifaceted challenges inherent in traditional examination administration. This project has not only revolutionized the way teachers, students, and administrators interact with examination-related tasks but has also paved the way for a culture of efficiency, transparency, and empowerment within our educational ecosystem. By centralizing exam content management, result dissemination, practice resources, and communication channels, the website has streamlined operations, reduced administrative burden, and enhanced accessibility for all stakeholders.

# **Bibliography**

- [1] Udemy- <https://www.udemy.com/course/nodejs-express-mongodb-bootcamp>
- [2] Google
- [3] Stackoverflow