**Java Full Stack Development: IamNeo and**

**Microsoft Azure Administrator Associate**

A REPORT

submitted by

### MEKALA MANIKANTA

### 21BEC0992

*in partial fulfilment for the award*

of

**B. Tech. Electronics and Communication Engineering**

**SENSE**



## April 2024



**School of Electronics and Communication Engineering**

# DECLARATION

I hereby declare that the project entitled “**JAVA FULL STACK DEVELOPMENT and MICROSOFT AZURE ADMINISTRATOR (AZ-104)**” submitted by me to the School of electronics and communication engneering, Vellore Institute of Technology, vellore Campus, vellore 632014 in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology – Electronics engineering and communication** is a record of bonafide work carried out by me**.** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Signature

**MEKALA MANIKANTA**

**21BEC0992**



**School of Electronics Communication and Engineering**

# CERTIFICATE

The project report entitled “**JAVA FULL STACK DEVELOPMENT and MICROSOFT AZURE ADMINISTRATOR (AZ-104)**” is prepared and submitted by **MEKALA MANIKANTA (Register No: 21BEC0992)**. It has been found satisfactory in terms of sense, quality and presentation as partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology – Electronics communication and Engineering** in Vellore Institute of Technology, Vellore, India.

**Examined by**:

**Examiner I Examiner II**

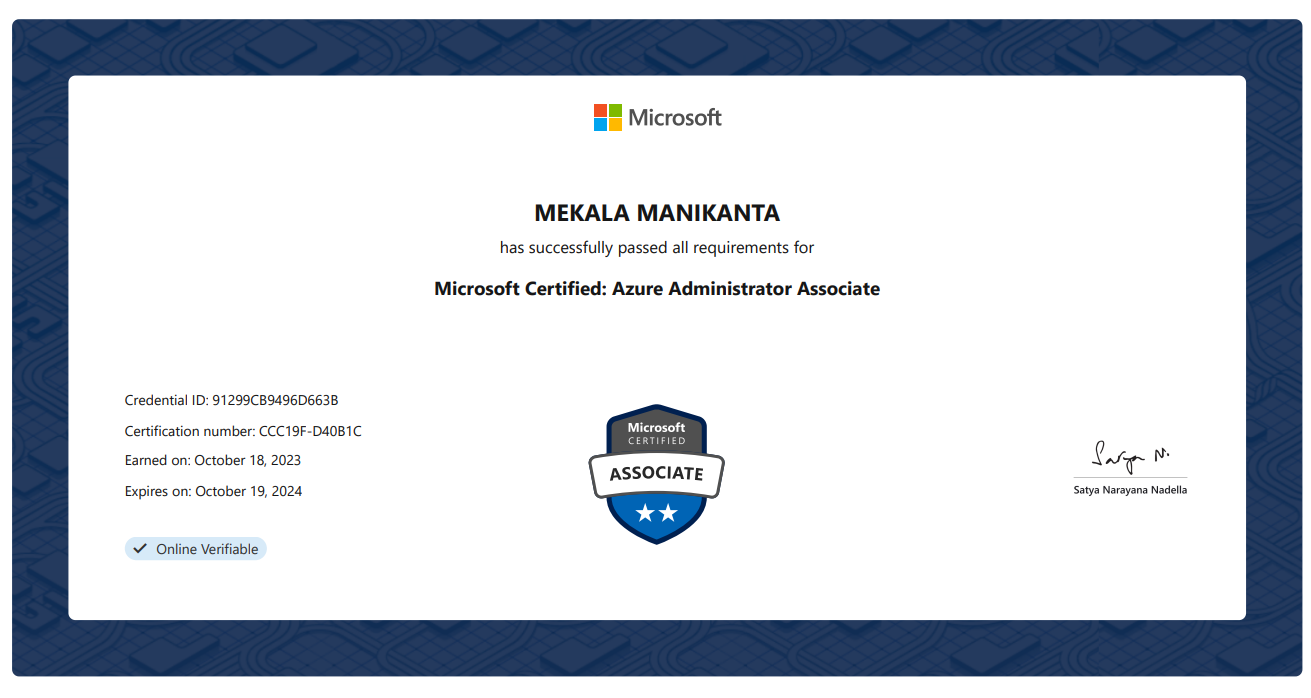
**Java Full Stack Development:**



**A certificate of completion

Description automatically generated**

**Microsoft Azure :**



# ACKNOWLEDGEMENT

As I present this project report, I express profound gratitude to the individuals and organizations whose unwavering support and guidance have enabled its completion. Throughout this challenging journey, their assistance and encouragement have been invaluable, shaping both my academic and personal growth.

I extend heartfelt thanks to my project supervisor, **Dr. Nithyanandam P**, Head of the Department (HoD) of B.Tech Electronics and Communication Engineering (ECE) at VIT Vellore. Their guidance, patience, and expertise have been instrumental, providing insightful feedback and constructive criticism that refined my work and fostered my development.

The faculty members and Deans of the School of Electronics and communication Engineering at VIT Vellore, particularly **Dr. Ganesan R**, **Dr. Parvathi R**, and **Dr. Geeta S**, deserve deep appreciation for their dedication to imparting knowledge and nurturing academic excellence.

My classmates and peers have been a constant source of support and encouragement, motivating me throughout this journey. Additionally, I owe a debt of gratitude to my family for their unwavering belief in my abilities, providing the strength and determination to pursue this project.

I am also thankful for the guidance provided by Haripriya R, a Full Stack Developer at IamNeo, whose insights significantly contributed to the project's development.

Lastly, I am grateful to VIT Vellore for providing the necessary resources and opportunities, facilitating invaluable learning experiences essential for future endeavors.

This project report symbolizes the collective effort and support of all involved. I deeply appreciate their contributions to this transformative journey. Thank you for your unwavering support and encouragement.

Place: Vellore Mekala Manikanta

Date: 21BEC0992

# CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Title** | **Page** |
|  | Title Page | 1 |
|  | Declaration | 2 |
|  | Certificate | 3 |
|  | Industry certificate | 4 |
|  | Acknowledgement | 5 |
|  | Table of contents | 6 |
|  | List of Figures | 7 |
|  | List of Abbreviation | 8 |
|  | Abstract | 9 |
| 1 | Introduction | 13 |
| 2 | Course Objective | 14 |
| 3 | Foundation Certificate: Java FullStack – IamNeo | 16 |
| 3.1 | HTML | 16 |
| 3.2 | CSS | 16 |
| 3.3 | JAVASCRIPT | 16 |
| 3.4 | RDBMS | 17 |

|  |  |  |
| --- | --- | --- |
| 3.5 | BASIC JAVA PROGRAMING | 17 |
| 3.6 | ADVANCED JAVA PROGRAMIING | 17 |
| 3.7 | WEB SERVICES | 18 |
| 3.8 | SPRING | 18 |
| 3.9 | JPA | 19 |
| 3.10 | TYPESCRIPT | 21 |
| 3.11 | ANGULAR JS | 21 |
| 3.12 | PROJECT | 22 |
| 4 | Global Certificate: Microsoft Azure Administrator | 24 |
| 4.1 | Manage Azure Identities and Governance | 25 |
| 4.2 | Implement and Manage Storage | 26 |
| 4.3 | Deploy and Manage Azure Compute Resources | 26 |
| 4.4 | Configure and Manage Virtual Networking | 27 |
| 4.5 | Monitor and Back Up Azure Resources | 27 |
| 5 | Conclusion | 29 |
|  | References | 30 |
|  | Appendix – I | 30 |

**LIST OF FIGURES**

**Title Page**

Fig 1 : Landing Page 30

31

Fig 2 : Adding student

Fig 3 : Student list 31

Fig 4 : Delete the student 32

Fig 5 : Spring Source code 32

Fig 6 : Angular Source code 33

# LIST OF ABBREVIATIONS

### Abbreviation Expansion

HTML Hypertext Markup Language CSS Cascading Style Sheets

Js Java Script

SQL Structed Query Language

JPA Java Persistence API

CORS CrossOrgin resource sharing

DOM Document Object Model

HTTP Hyper Text Transfer Protocol IoC Inversion of Control

CRUD Create, Read, Update, Delete Operations

REST API Representational State Transfer Application Programming Interface JVM Java Virtual Machine

JRE Java Runtime Environment

JDK Java Development Kit

# ABSTRACT

The internship courses in JAVA Full Stack Development and Azure Administrator Associate have been instrumental in providing us with a holistic understanding of two critical domains in the realm of technology. The JAVA Full Stack Development course, spanning over two months, delved deep into essential web development technologies such as HTML, CSS, JavaScript, RDBMS, Basic and Advance Java Programming, Spring, Java Persistence API (JPA), TypeScript and Angular.js. Through a series of meticulously crafted classes and hands-on exercises, we were equipped with the knowledge and skills necessary to design, develop, and deploy full-stack web applications. The course culminated in a capstone project to create an Student Management System, demonstrating their proficiency in applying the concepts learned to real-world scenarios.

The goal of the Microsoft Azure Administrator (AZ-104) certification program is to give IT workers the abilities they need to efficiently administer Azure resources. This course covers a wide range of topics, including building virtual networks, managing Azure identities and governance, installing and managing Azure compute resources, implementing and managing storage solutions, and monitoring and backing up Azure resources. Participants acquire the skills necessary to effectively manage Azure infrastructures, guaranteeing seamless operations and peak performance, through interactive laboratories and realistic exercises. Administrators may handle a variety of Azure jobs, such as data storage optimization and user access control, by learning the AZ-104 curriculum.

The proposed solution consists of many main components designed to meet the highlighted difficulties. First, the **application's user interface** will be improved to make it more intuitive and user-friendly. This involves improving elements like the navigation bar, student list, student addition form, and student information page. Second, **backend integration** will be created to link the frontend application to the current Spring Boot backend, allowing for easy data interchange and manipulation. This integration will need creating API endpoints to handle student data retrieval, insertion, and update. In addition, steps will be taken to guarantee adequate CORS setup for safe cross-origin communication.

In conclusion, the internship courses in Java Full Stack Development and Azure Administrator Associate have played a pivotal role in equipping us with the knowledge, skills, and experience needed to excel in the dynamic and ever-evolving field of technology. As we embark on their professional journeys, we carry with them a solid foundation in both web development and cloud Platforms, positioning them for success in a wide range of career opportunities and empowering them to make meaningful contributions to the digital transformation of industries worldwide.

# 1 INTRODUCTION

In the contemporary landscape of technology-driven advancements, staying abreast of emerging trends and acquiring practical skills is imperative for individuals seeking to excel in today's competitive job market. To address this need, our educational institution has curated a comprehensive series of internship courses, encompassing two pivotal domains of technology: Java Full Stack Development and Microsoft Azure Administrator AZ-104.

The Java Full Stack Development course immerses us into the realm of web development, offering a deep dive into essential technologies including HTML, CSS, JavaScript, RDBMS, Basic and Advance Java Programming, Spring, Java Persistence API (JPA), TypeScript and Angular.js. Over the span of two months, we undergo rigorous training and hands-on exercises to master the art of building robust and interactive web applications. The course culminates in a collaborative project where we develop a Student Management System, showcasing their proficiency in full-stack web development and practical problem-solving skills.

Simultaneously, Microsoft Azure Administrator (AZ-104) course gives IT workers the abilities they need to efficiently administer Azure resources. This course covers a wide range of topics, including building virtual networks, managing Azure identities and governance, installing and managing Azure compute resources, implementing and managing storage solutions, and monitoring and backing up Azure resources. Participants acquire the skills necessary to effectively manage Azure infrastructures, guaranteeing seamless operations and peak performance, through interactive laboratories and realistic exercises. Administrators may handle a variety of Azure jobs, such as data storage optimization and user access control, by learning the AZ-104 curriculum.

While distinct in their focus, both internship courses share a common objective: to equip us with the knowledge, skills, and practical experience necessary to thrive in today's technology-driven landscape. By providing a comprehensive understanding of web development and cloud computing fundamentals, these courses prepare us for a myriad of career opportunities in fields ranging from software engineering to cloud architecture. Moreover, the collaborative and project- based approach fosters teamwork, creativity, and critical thinking skills, enabling us to navigate the complexities of the digital age with confidence and competence.

As we embark on their professional journeys, the knowledge, skills, and experience gained from these internship courses serve as a solid foundation, empowering them to make meaningful contributions to the ever-evolving field of technology. With a focus on practical application and real-world problem-solving, these courses underscore our commitment to equipping ourselves with the tools We need to succeed in today's dynamic and competitive job market.

# 2 COURSE OBJECTIVE

The Java Full Stack Development course provides an immersive experience in web development, covering essential technologies including HTML, CSS, JavaScript, relational databases, Basic and Advanced Java Programming, Spring framework, Java Persistence API (JPA), TypeScript, and Angular.js. Over a two-month duration, participants undergo rigorous training and hands-on exercises to master the art of building robust and interactive web applications. The course concludes with a collaborative project where participants develop a Student Management System, demonstrating proficiency in full-stack web development and practical problem-solving skills.

Simultaneously, the Microsoft Azure Administrator (AZ-104) course equips IT professionals with the skills needed to efficiently administer Azure resources. Covering a wide range of topics, including building virtual networks, managing Azure identities and governance, deploying and managing Azure compute resources, implementing and managing storage solutions, and monitoring and backing up Azure resources. Participants gain practical skills necessary to effectively manage Azure infrastructures through interactive laboratories and realistic exercises. By completing the AZ-104 curriculum, administrators are equipped to handle various Azure tasks, including data storage optimization and user access control, ensuring seamless operations and peak performance.

# 3 JAVA FULL STACK DEVELOPMENT

The Java Full Stack Development course covers a range of topics essential for web development, including HTML, CSS, JavaScript, RDBMS, Basic and Advance Java Programming, Spring, Java Persistence API (JPA), TypeScript and Angular.js. The course is divided into modules, with each module focusing on a specific technology or concept. We will engage in lectures, Assignments, and hands-on projects to reinforce their learning and apply their skills in real-world scenarios. The culmination of the course is a capstone project to develop an Student Management System, integrating their knowledge and skills acquired throughout the course.

# HTML

Course introduces us to the fundamental building blocks of web development. Through hands- on exercises and projects, we learn to create structured and semantically meaningful web pages using HTML elements and tags. We gain proficiency in designing forms, input fields, and other interactive elements, enhancing user experience and accessibility.

# CSS

We delve into the world of cascading style sheets, mastering the art of styling and formatting web content. We learn to select and apply CSS properties to HTML elements, customise layouts, and create visually appealing designs. Through practical labs and projects, we gain expertise in CSS layouts, including flexbox and grid systems, enabling them to create responsive and dynamic web interfaces. Furthermore, we learn to leverage CSS frameworks like Bootstrap to streamline development and ensure consistency across different browsers and devices.

## Java Script

We dive into the dynamic programming language that powers interactive web pages and web applications. We learn JavaScript syntax, data types, and control structures, enabling them to manipulate the DOM and handle user interactions. Through practical exercises and projects, we gain proficiency in writing JavaScript functions, managing scope, and implementing event-

driven programming. We also explore modern JavaScript features like arrow functions, destructuring, enhancing code readability and maintainability.

# RDBMS

In Mastering RDBMS involves understanding DDL/DML for database structure and manipulation. Subqueries and aggregate functions aid in complex data analysis. Joins integrate data from multiple tables for comprehensive insights. Leveraging frameworks like MySQL and PostgreSQL streamlines workflow. Database administrators ensure optimal performance and data integrity. RDBMS mastery empowers informed decision-making and innovation.

## Basic Java Programming

In mastering basic Java programming, we delve into fundamental concepts such as JDK, JRE, and JVM. Understanding the Java Development Kit (JDK) equips us with tools for writing and compiling Java code, while the Java Runtime Environment (JRE) provides an environment for executing Java applications. The Java Virtual Machine (JVM) serves as the platform for running Java bytecode on different operating systems. Additionally, we explore essential programming constructs like arrays for storing and manipulating data efficiently. Embracing object-oriented programming (OOP) concepts such as classes, objects, inheritance, polymorphism, and encapsulation enables us to design modular, reusable, and scalable Java applications.

## Advanced Java Programming

We learn a wide range of important topics and techniques in advanced Java programming. We look at several data structures like as linked lists, arrays, and queues, as well as collections like ArrayLists, HashMaps, and Sets, to help our programmes manage data more efficiently. Understanding access specifiers like as public, private, protected, and default enables us to regulate the visibility and accessibility of classes, methods, and variables while maintaining code integrity and security. Interfaces serve as class blueprints, allowing for code reuse and polymorphism, whilst Comparator and Comparable interfaces give versatile sorting algorithms for items in collections. Exception handling methods, like as try-catch blocks and throw

statements, allow us to handle runtime mistakes gracefully, improving programme stability. Furthermore, file handling features provide us with the ability to communicate with external resources, allowing for data persistence and modification using techniques such as file input/output streams and readers/writers. These fundamental principles and technologies serve as the foundation for Java full-stack development, allowing us to create strong, scalable, and dependable systems that are customised to current software requirements.

## Web Services

We begin with an overview of web services, which are software systems designed to enable interoperable machine-to-machine communication across a network. Servlets, Java-based components that operate on a web server, are essential for managing client requests and creating dynamic online content. A thorough grasp of the HTTP protocol, which serves as the foundation for data exchange on the World Wide Web, is also required. These components work together to provide the backbone of Java web development, allowing you to create powerful and scalable applications that meet current software standards.

## Spring

In Java full-stack development, the Spring framework serves as a cornerstone, providing a comprehensive suite of tools and functionalities. At its core, Spring's Inversion of Control (IoC) container revolutionizes application development by managing object creation and lifecycle, promoting loose coupling and easy integration. Autowiring further enhances development efficiency by automating dependency injection, reducing configuration overhead through annotations like **@Autowired** and **@Component**.

With Spring's support for RESTful services, developers can effortlessly create CRUD operations and REST APIs using annotations like **@RestController**, facilitating seamless communication between clients and servers. Additionally, Spring's layered architecture promotes modularization, ensuring clear separation of concerns and enabling scalability and maintainability. Robust HTTP response code handling enhances the reliability of web applications, providing clear communication with clients during various scenarios.

Integration with Object-Relational Mapping (ORM) frameworks like Hibernate or JPA simplifies database interactions, abstracting away SQL intricacies and enabling rapid development. Leveraging Spring's seamless integration with Hibernate or JPA annotations like **@Entity** and **@Repository** streamlines data access and management, reducing boilerplate code and enhancing productivity. Collectively, these Spring components empower developers to build robust, scalable, and maintainable Java applications, making Spring a go-to choose for modern software development projects. JPA annotations like **@Entity**, **@Table**, **@Id**, and **@GeneratedValue** define the mapping between Java objects and database tables, streamlining data persistence.

# JPA

1. @Entity:
   * Annotation used to mark a Java class as an entity, indicating that instances of this class will be managed by the JPA provider and mapped to database tables.
   * Example: @Entity
2. @Table:
   * Annotation used to specify the details of the database table to which the entity is mapped.
   * Allows customization of table name, schema, and other properties.
   * Example: @Table (name = "users")
3. @Id:
   * Annotation used to specify the primary key attribute of an entity.
   * Marks the field or property as the unique identifier for instances of the entity.
   * Example: @Id
4. @GeneratedValue:
   * Annotation used to specify how the primary key value should be automatically generated.
   * Supports strategies like AUTO, IDENTITY, SEQUENCE, and TABLE.
   * Example: @GeneratedValue (strategy = GenerationType.IDENTITY)
5. @Column:
   * Annotation used to specify the mapping of an entity attribute to a database column.
   * Allows customization of column name, type, length, and other properties.
   * Example: @Column(name = "username", length = 50)
6. @OneToMany:
   * Annotation used to define a one-to-many relationship between two entities.
   * Indicates that one instance of the source entity can be associated with multiple instances of the target entity.
   * Example: @OneToMany(mappedBy = "user")
7. @ManyToOne:
   * Annotation used to define a many-to-one relationship between two entities.
   * Specifies that multiple instances of the source entity can be associated with a single instance of the target entity.
   * Example: @ManyToOne
8. EntityManager.persist():
   * Method used to persist a new entity instance into the database.
   * Adds the entity object to the persistence context and schedules an insert operation upon transaction commit.
   * Example: entityManager.persist(user)
9. EntityManager.find():
   * Method used to retrieve an entity instance from the database based on its primary key.
   * Searches for the entity with the specified primary key value and returns the corresponding object.
   * Example: User user = entityManager.find(User.class, 1L)
10. EntityManager.createQuery():
    * Method used to create a JPQL query.
    * Accepts a JPQL string as input and returns a Query object that can be executed to retrieve data from the database.
    * Example: Query query = entityManager.createQuery("SELECT u FROM User u WHERE u.age > 18")

These commands and annotations form the backbone of JPA, providing developers with powerful tools to map Java objects to relational database tables and perform CRUD operations with ease.

## TypeScript

TypeScript, a superset of JavaScript, enhances code quality through optional static typing and advanced features like interfaces and enums. It differs from JavaScript by introducing static typing and interfaces for defining object shapes. Arrays in TypeScript provide type safety, allowing developers to define arrays with specific element types. Promises in TypeScript streamline asynchronous operations, enabling cleaner and more readable code with improved error handling using .then() and .catch() . This combination of features makes TypeScript a powerful tool for building robust and maintainable JavaScript applications, especially in modern web development workflows.

## Angular JS

Angular serves as a cornerstone for crafting dynamic and interactive web applications. Its component-based architecture promotes modularity and reusability, facilitating encapsulation of UI elements and logic. Angular's powerful data binding mechanisms enable seamless synchronization between the component's view and model, including two-way data binding for instant updates. With pipes, developers can transform and format data within templates, enhancing user experience. Directives extend HTML with custom behaviour, empowering developers to manipulate the DOM dynamically. Angular's built-in router facilitates navigation between views, including handling page-not-found scenarios for a smooth user journey. Form control abstraction

simplifies form management, while reactive forms offer a flexible approach to form development with comprehensive validation support. Custom async validators and error handling ensure data integrity and user feedback. Angular's testing ecosystem, including Jasmine and Karma, enables rigorous unit and integration testing, ensuring code quality and reliability. Mastering Angular empowers Java full-stack developers to create modern, scalable, and maintainable web applications that meet the demands of today's dynamic digital landscape.

## Project

The Student Management System for ABC University aims to modernize and streamline administrative processes within the institution. By integrating Angular for the frontend and Spring Boot for the backend, the system enhances user experience and data management capabilities. The frontend improvements include creating components like Navbar, Student List, Add Student, and Student Details, each with its respective functionality and CSS styling. Meanwhile, the backend is structured with components such as ApiController, Student Model, StudentRepo, StudentService, and CORS Configuration to handle API endpoints for adding students, retrieving student details, and fetching student lists.

This project tackles the challenges faced by ABC University's existing student management system by providing a comprehensive solution that improves data organization, accessibility, and user interaction. The frontend enhancements ensure a more intuitive interface for users to navigate and interact with student data, while the backend integration facilitates seamless communication between the frontend and the database. By following the specified folder structure, naming conventions, and API endpoint guidelines, the system ensures consistency and efficiency in development and maintenance. Ultimately, this project empowers ABC University to better manage its student information and optimize administrative workflows for enhanced productivity and effectiveness.

API Endpoints:

1. POST /addStudent:

Description: Adds a new student to the backend database. Request Body:

JSON Object:

* + id (int): Registration ID of the student.
  + name (String): Name of the student.
  + department (String): Department of the student.
  + phonenumber (String): Phone number of the student. Response: Boolean (true if successful, false if not)

1. GET /getById/{id}:

Description: Fetches a specific student's details based on the provided ID. Path Variable: id (int): The unique ID of the student to fetch.

Response: JSON Object (Student object)

1. GET /getAllStudent:

Description: Fetches a list of all students from the backend. Response: JSON Array (List of Student objects)

# 4 MICROSOFT AZURE DEVELOPMENT

A wide range of operating systems, programming languages, frameworks, tools, databases, and devices are supported by Azure, a public cloud service platform. It can create apps with JavaScript, Python, NET, PHP, Java, and Node.js; it can run Linux containers with Docker integration; and it can create back-ends for iOS, Android, and Windows devices.

The same technologies that millions of developers and IT professionals now rely on and trust are supported by Azure public cloud services. When you expand your IT infrastructure or move your data to a public cloud service provider, you are depending on that company's ability to secure your data and apps using the services and security controls they offer to handle the security of your cloud-based assets.

Azure's infrastructure offers a reliable base on which companies may meet their security needs. It is built from the ground up to accommodate millions of clients at once, from facilities to applications.

## AZ-104 Pattern:

The AZ-104 exam focuses on common tasks and concepts that an administrator needs to understand to deploy and manage infrastructure in Microsoft Azure. Manage Azure identities and Azure subscriptions is a key topic on the exam, which includes managing Azure AD objects (users, groups, and devices), use of Azure AD join and self-service password resets; it also covers role- based access control, tagging, subscription level policies and resource organization using resource groups, subscription and management groups. Another topic covered is implement and manage storage, which includes creating and configuring storage accounts as well as configuring Azure files and understanding the services for importing and exporting data to Azure. A significant portion of the exam is focused on deploying and managing Azure compute resources, which includes configuring high availability of Azure VMs, creating and configuring virtual machine and their automated deployments as well as creating and configuring container solutions such as Azure Kubernetes Service (AKS) and Azure Container Instances (ACI); it also covers configuring web apps using app service and app service plans. This book also covers the creation and management of virtual networks, DNS, connectivity between virtual networks, configuring network security

groups, Azure firewall and Azure bastion service; it also explains the load balancing solutions including configuration of application gateway. The final topic is monitor and backup Azure resources, which includes topics on how to monitor resources using Azure Monitor as well as how to implement back and recovery of Azure VMs including site to site recovery using Azure site recovery.

## Manage Azure Identities and Governance

Managing Azure identities and governance involves implementing robust identity management solutions and establishing governance frameworks to ensure compliance, security, and efficient resource management within Azure environments. This includes:

1. **Azure Active Directory (Azure AD):** Implement and manage user identities, groups, and roles using Azure AD. Utilize features like Multi-Factor Authentication (MFA), Conditional Access policies, and Identity Protection to enhance security.
2. Role-Based Access Control (RBAC): Define and assign roles to users or groups based on their responsibilities within Azure. Use RBAC to control access to Azure resources and enforce least privilege access principles.
3. Azure Policy: Define and enforce policies to govern resource configurations and compliance requirements across Azure subscriptions. Implement policies for resource tagging, naming conventions, security controls, and regulatory compliance.
4. Management Groups: Organize Azure subscriptions into logical groups called management groups to apply governance controls and policies at scale. Use management groups to centrally manage access, compliance, and billing for multiple subscriptions.
5. Cost Management: Monitor and optimize Azure spending by implementing cost management practices such as budgeting, cost analysis, and resource tagging. Utilize Azure Cost Management + Billing to track and manage Azure expenditures effectively.

## Implement and Manage Storage

Implementing and managing storage in Azure involves provisioning, configuring, and securing storage resources to store and manage data efficiently. Key aspects include:

1. Azure Storage Accounts: Create and configure Azure storage accounts to store various types of data, including blobs, files, tables, and queues. Implement access controls, encryption, and replication options to ensure data security and availability.
2. Data Protection: Implement data protection measures such as encryption at rest and in transit, access controls, and backups to safeguard data stored in Azure Storage. Utilize features like Azure Key Vault for managing encryption keys securely.
3. Storage Tiers and Lifecycle Management: Optimize storage costs by leveraging storage tiers (hot, cool, archive) based on data access patterns and implementing lifecycle management policies to automate data tiering and retention.
4. Azure Files and Azure Blob Storage: Configure Azure Files for cloud-based file shares and Azure Blob Storage for scalable object storage. Implement features like Azure File Sync for hybrid cloud file sharing and Blob Versioning for data protection.

## Deploy and Manage Azure Compute Resources

Deploying and managing Azure compute resources involves provisioning, configuring, and managing virtual machines, containers, and serverless computing resources. Key tasks include:

1. Virtual Machines: Deploy and manage virtual machines (VMs) in Azure to run applications and workloads. Configure VM sizes, operating systems, networking, and storage options based on workload requirements.
2. Azure Kubernetes Service (AKS): Deploy and manage Kubernetes clusters using AKS to orchestrate containerized applications at scale. Use AKS features for automatic scaling, monitoring, and high availability.
3. Azure App Service: Deploy and manage web applications, APIs, and mobile backends using Azure App Service. Utilize features like deployment slots, auto-scaling, and continuous integration/continuous deployment (CI/CD) for efficient application management.
4. Serverless Computing: Implement serverless computing solutions using Azure Functions and Logic Apps to build event-driven, scalable, and cost-effective applications. Utilize triggers, bindings, and monitoring capabilities to automate workflows and processes.

## Configure and Manage Virtual Networking

Configuring and managing virtual networking in Azure involves designing, implementing, and securing network infrastructure to enable communication between Azure resources, on-premises networks, and the internet. Key aspects include:

1. Azure Virtual Network (VNet): Create and configure VNets to isolate and segment Azure resources logically. Define subnets, IP address ranges, and network security groups (NSGs) to control traffic flow and enforce security policies.
2. Connectivity Options: Establish secure connections between Azure VNets, on-premises networks, and external networks using Azure VPN Gateway, Azure ExpressRoute, or Azure Virtual WAN. Configure VPN and ExpressRoute gateways, peering, and routing to enable connectivity.
3. Network Security: Implement network security controls such as NSGs, application security groups (ASGs), and Azure Firewall to filter and control inbound and outbound traffic. Monitor network traffic and configure logging and alerts for security incidents.
4. Azure Bastion: Securely access Azure VMs using Azure Bastion, a fully managed platform-as-a-service (PaaS) solution for secure RDP and SSH connectivity to VMs without exposing public IP addresses.

## Monitor and Back Up Azure Resources

Monitoring and backing up Azure resources involves implementing monitoring solutions to track resource performance, availability, and security, as well as implementing backup and recovery solutions to protect data and applications from loss or corruption. Key activities include:

1. Azure Monitor: Set up and configure Azure Monitor to collect and analyze metrics, logs, and diagnostic data from Azure resources. Create alerts, dashboards, and visualizations to monitor resource health, performance, and security.
2. Backup and Recovery: Establish backup and recovery strategies using Azure Backup, Azure Site Recovery (ASR), and Azure SQL Database backups. Configure backup

policies, retention periods, and recovery plans to protect data and applications from disasters and outages.

1. Disaster Recovery: Implement disaster recovery solutions using ASR to replicate VMs and applications to a secondary Azure region for failover in case of an outage. Test failover scenarios regularly to ensure business continuity and data integrity.
2. Security Monitoring: Enable security monitoring and threat detection using Azure Security Center to identify and mitigate security risks and compliance issues across Azure resources. Utilize built-in security policies, recommendations, and threat intelligence for proactive security management.
3. Continuous Improvement: Continuously monitor and optimize Azure resources, configurations, and costs using tools like Azure Advisor, Cost Management + Billing, and Azure Automation. Implement best practices, fine-tune configurations, and adjust resource allocations to improve efficiency and reduce costs over time.

# 5 CONCLUSIONS

The internship courses in Java Full Stack Development and Azure Administrator Associate have provided us with a comprehensive education encompassing two critical domains in the field of technology. Through these courses, we have gained practical skills, theoretical knowledge, and hands-on experience that are invaluable for success in today's dynamic job market.

The JAVA Full Stack Development course equipped us with proficiency in essential web development technologies, enabling them to design, develop, and deploy full-stack web applications. Through a combination of lectures, assignments, and projects, we honed their skills in HTML, CSS, JavaScript, RDBMS, Basic and Advance Java Programming, Spring, Java Persistence API (JPA), TypeScript and Angular.js, The culminating project, the Student Management System, provided an opportunity for us to apply their knowledge and collaborate with their peers, showcasing their ability to solve real-world problems using modern web development tools and techniques.

In conclusion, the Azure Administrator Associate certification exam, AZ-104, evaluates proficiency in various critical areas of Azure administration, including governance, identity management, storage, compute, networking, monitoring, and backup. Successfully passing this exam demonstrates the ability to effectively deploy, manage, and secure Azure environments, ensuring optimal performance, reliability, and cost-efficiency. By mastering the skills assessed in AZ-104, candidates prove their readiness to handle real-world challenges in cloud administration roles, making them valuable assets to organizations leveraging Azure for their infrastructure and applications. Furthermore, achieving Azure Administrator Associate certification opens up a world of opportunities for career advancement in the rapidly growing field of cloud computing, empowering individuals to contribute to the digital transformation and innovation of businesses worldwide.

The interdisciplinary nature of these internship courses has empowered us with a diverse skill set that is highly sought after in today's technology-driven landscape. By combining expertise in full- stack web development with proficiency in cloud computing, we are well-equipped to tackle complex challenges and drive innovation in their respective fields. Moreover, the collaborative and project-based approach fostered a sense of teamwork, creativity, and critical thinking among us, preparing them for success in collaborative work environments.

# REFERENCES

1. https://[www.examtopics.com/exams/microsoft/az-104/view/](http://www.examtopics.com/exams/microsoft/az-104/view/)
2. https://[www.geeksforgeeks.org/best-full-stack-project-ideas/](http://www.geeksforgeeks.org/best-full-stack-project-ideas/)
3. https://github.com/topics/student-management-system?l=java
4. https://learn.microsoft.com/en-us/credentials/certifications/resources/study-guides/az-104
5. https://[www.whizlabs.com/blog/az-104-microsoft-azure-administrator-certification/](http://www.whizlabs.com/blog/az-104-microsoft-azure-administrator-certification/)

# APPENDIX I

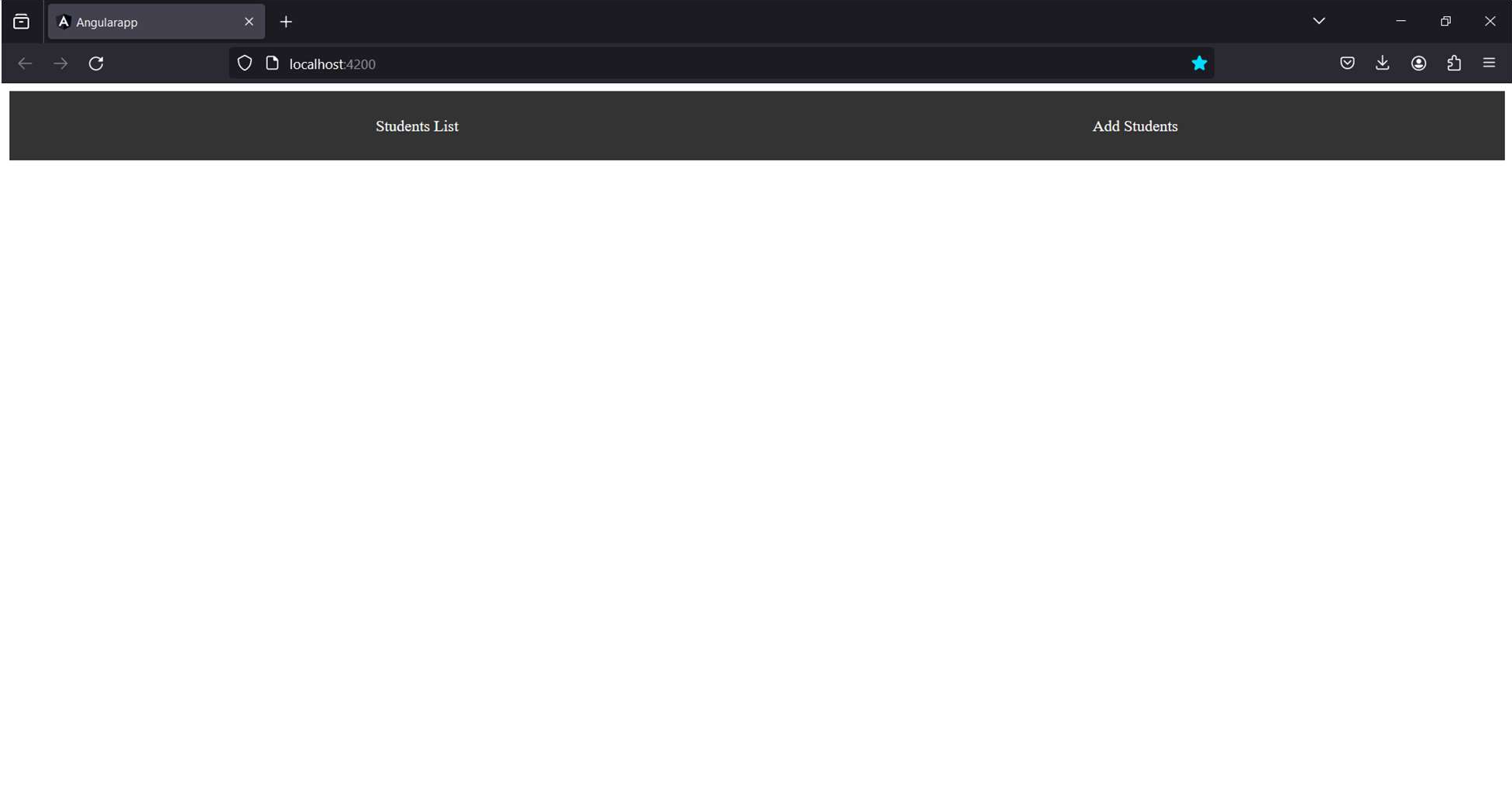


Fig 1: Landing Page

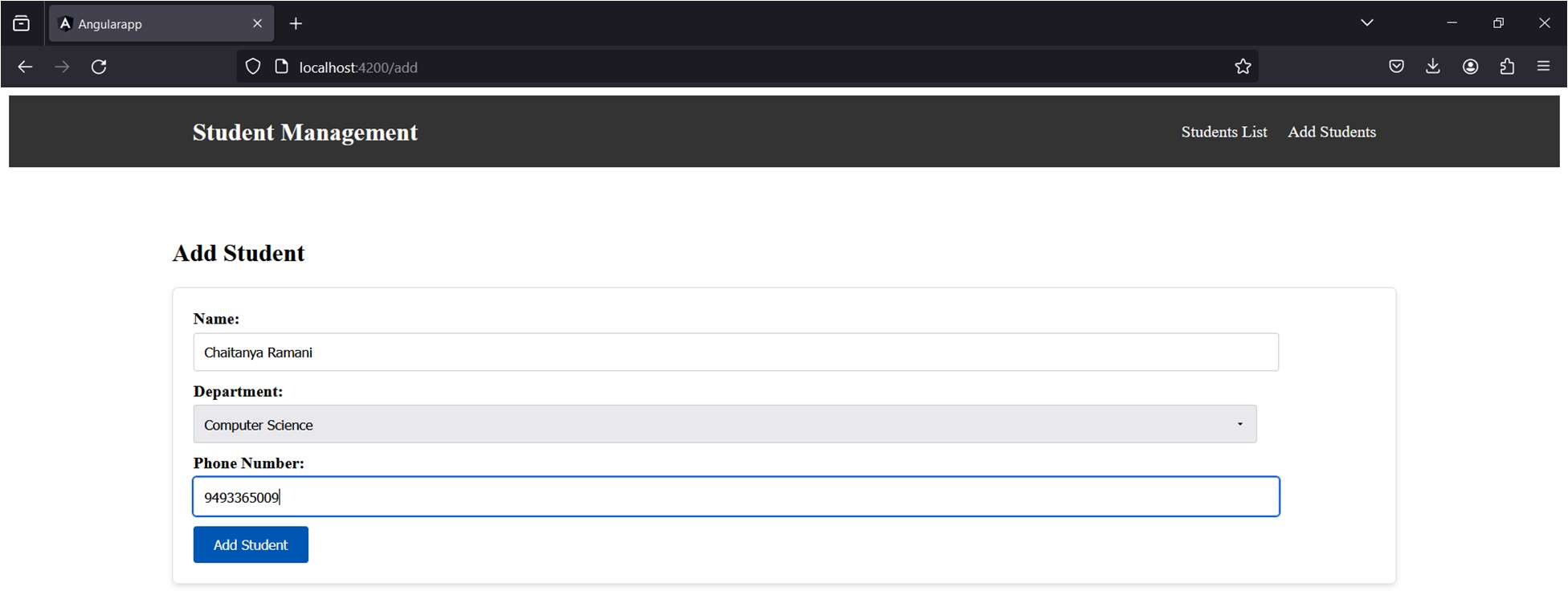


Fig 2: Adding Student

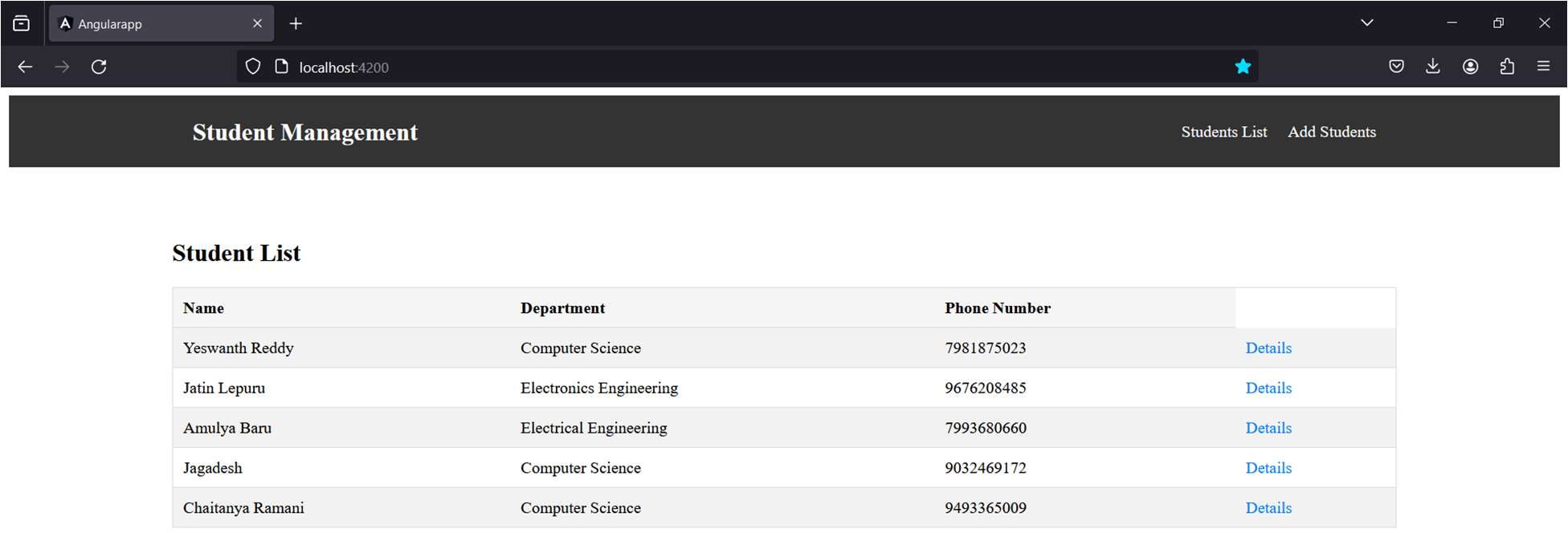


Fig 3: Students List

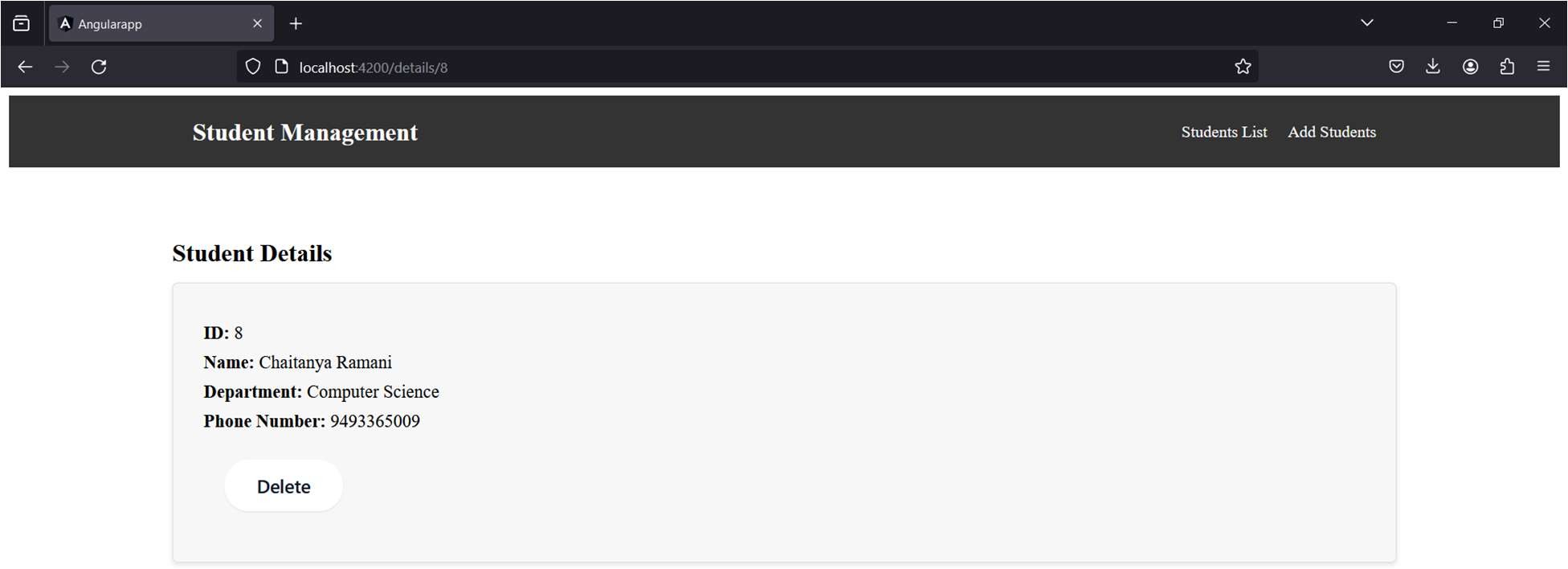


Fig 4: Delete of the Student

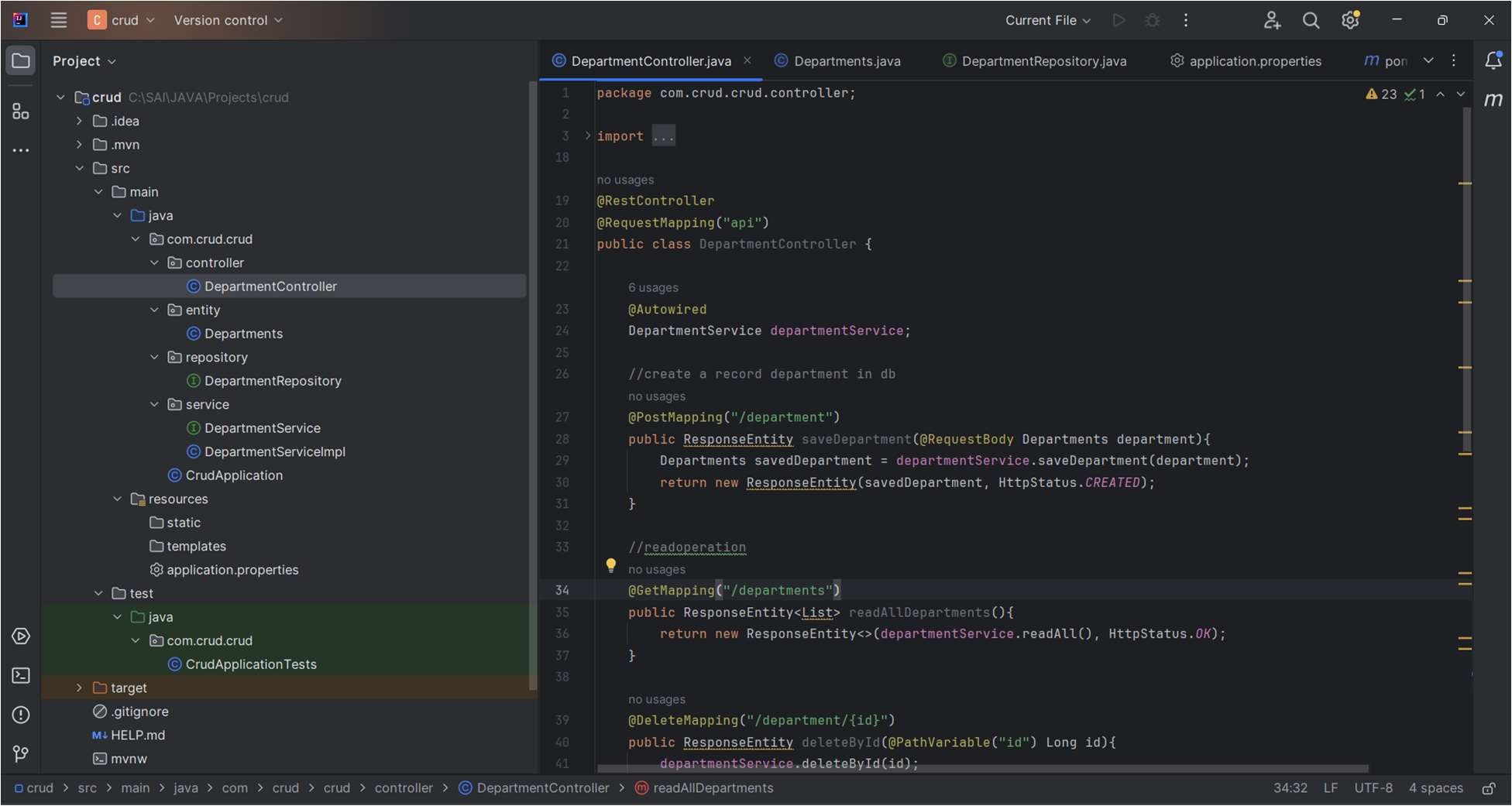


Fig 5: Spring Source Code

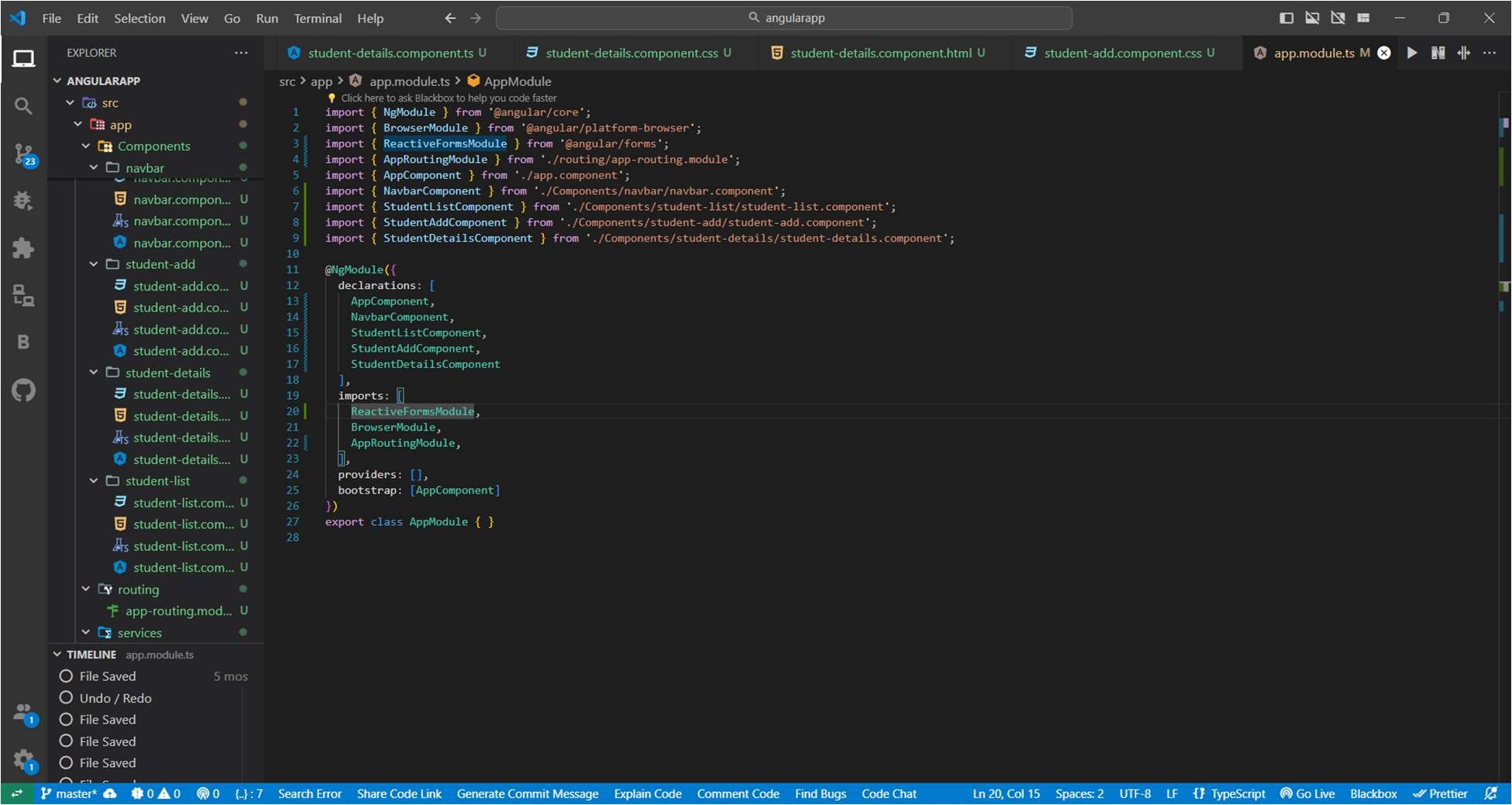


Fig 6: Angular Source Code