A PROJECT REPORT

on

STUDENT DATABASE MANAGEMENT SYSTEM

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

AKASH KUMAR JAISHWAL	21052935
GEETIKA PADAM	2105460

UNDER THE GUIDANCE OF
KAMLESH KARMAKAR

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

April 2024

A PROJECT REPORT

on

STUDENT DATABASE MANAGEMENT SYSTEM

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

BY

AKASH KUMAR JAISHWAL	21052935
GEETIKA PADAM	2105460

UNDER THE GUIDANCE OF

KAMLESH KARMAKAR

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certified that the project entitled

STUDENT DATABASE MANAGEMENT SYSTEM

submitted by

Akash Kumar Jaishwal	21052935
Geetika Padam	2105460

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering Computer Science & Engineering at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024, under our guidance.

Date: / /

Kamlesh Karmakar
Project Guide

Acknowledgements

We are profoundly grateful to **KAMLESH KARMAKAR** of **KIIT University** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

AKASH KUMAR JAISHWAL
GEETIKA PADAM

ABSTRACT

The "Student Database Management System" is a collaborative project undertaken by Akash Kumar Jaishwal and Geetika Padam, students of Computer Science and Engineering at KIIT University, under the guidance of Assistant Professor Kamalesh Karmakar. This project is developed as part of the coursework in cloud computing and aims to leverage cloud technologies to streamline student information management processes in educational institutions. The system offers an innovative approach to educational administration by integrating cloud computing principles into student data management. Through diligent research and practical implementation, the project seeks to optimize data management processes and enhance the efficiency of educational institutions in managing student information. Key features of the system include efficient storage and retrieval of student data, seamless integration with cloud-based services, and scalability to accommodate the evolving needs of educational institutions. The project underscores the commitment of the developers to academic excellence and innovation in the field of computer science, while also serving as a practical demonstration of the potential of cloud computing in educational administration.

Keywords: Student Database Management System, Cloud Computing, Amazon Web Service (AWS), Educational Administration, Data Management, Scalability, Efficiency, Academic Excellence, Innovation, Educational Institutions

Contents

List of Figures

Chapter 1

Introduction

As we navigate the digital age, the adept handling of student records becomes crucial for educational entities to optimize administrative operations and bolster learning outcomes. Traditional approaches to student data management often fall short, necessitating a more advanced solution. Enter the Student Database Management System (SDMS), a pivotal innovation that provides a unified and efficient platform for the storage, retrieval, and administration of student information.

This initiative seeks to bridge the deficiencies observed in existing solutions by harnessing the power of cloud computing to forge a resilient and scalable SDMS. The shift from manual or antiquated database systems to a cloud-based framework allows educational institutions to surmount challenges pertaining to data access, security, and expandability. Moreover, the SDMS promotes smooth collaboration among all parties involved administrators, educators, and students alike thereby nurturing an environment conducive to scholastic distinction.

The ensuing report will unfold with a synopsis of the project's goals and methods, followed by a comprehensive examination of the system's design and deployment specifics. It will further explore the advantages of embracing the SDMS, such as fortified data protection, augmented data reachability, and adaptability. The discourse will culminate with contemplations on the project's prospective trajectory and suggestions for incremental improvements to align with the dynamic demands of the educational landscape.

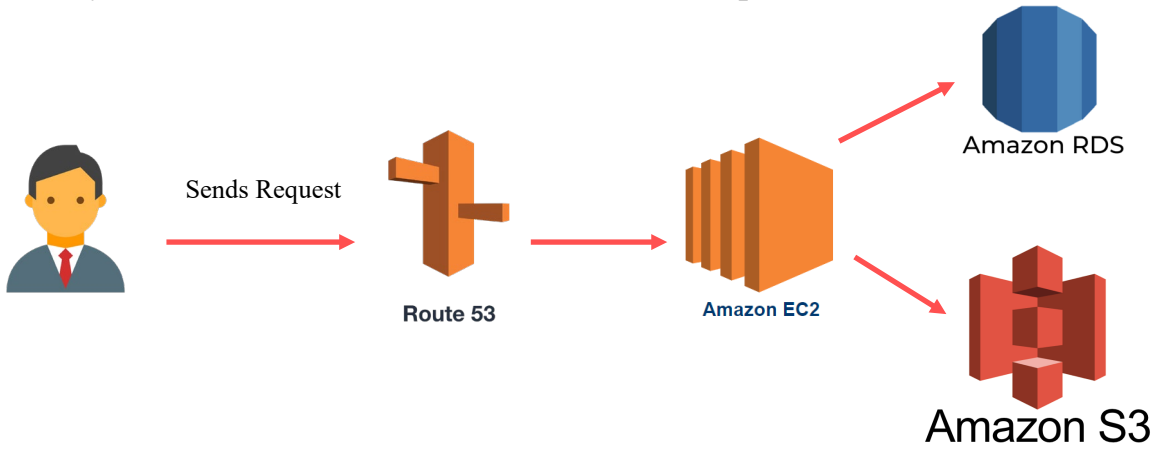

Fig 1.1: Working Diagram of SDMS

Chapter 2

Literature Review

1. Title: "Cloud Computing Security: A Comprehensive Survey"

- Authors: Siani Pearson and Yogesh Simmhan
- Summary: This survey paper provides an overview of security issues and challenges in cloud computing environments. It discusses various security threats, such as data breaches, unauthorized access, and insider attacks, and examines existing security mechanisms and best practices for mitigating these risks. The paper also highlights the importance of encryption, access control, and authentication in ensuring the security and privacy of cloud-based systems.

2. Title: "Database Management Systems: A Comprehensive Review"

- Authors: Raghu Ramakrishnan and Johannes Gehrke
- Summary: Ramakrishnan and Gehrke's review paper presents a comprehensive overview of database management systems (DBMS) and their underlying principles. It covers topics such as data models, query languages, transaction management, and database architectures, providing insights into the evolution of DBMS technology and its applications in various domains. The paper also discusses recent trends in database research, including NoSQL databases, big data analytics, and distributed database systems.

3. Title: "Python Programming Language: A Review of Features and Applications"

- Authors: Guido van Rossum and Barry Warsaw
- Summary: This review article examines the features and applications of the Python programming language, highlighting its simplicity, readability, and versatility. It discusses Python's syntax, data structures, and standard library modules, as well as its support for multiple programming paradigms. The paper explores Python's use cases in web development, scientific computing, data analysis, artificial intelligence, and machine learning, showcasing its widespread adoption and popularity among developers.

4. Title: "Flask: A Lightweight Web Framework for Python"

- Authors: Armin Ronacher and David Lord
- Summary: Ronacher and Lord's paper provides an in-depth analysis of Flask, a lightweight web framework for Python, focusing on its design principles, features, and use cases. It discusses Flask's minimalist architecture, which emphasizes simplicity and extensibility, and its support for essential web development tasks such as routing, request handling, and templating. The paper also examines Flask's ecosystem of extensions and its integration with other Python libraries, highlighting its flexibility and suitability for building a wide range of web applications.

5. Title: "AWS Cloud Development: A Review of Boto3 Library"

- Authors: Mitch Garnaat and Anton Efimenko
- Summary: This review paper evaluates Boto3, the Amazon Web Services (AWS) SDK for Python, focusing on its capabilities, usability, and integration with AWS services. It explores Boto3's object-oriented interface, which provides intuitive access to AWS resources and APIs, and its support for common cloud development tasks such as provisioning infrastructure, managing storage, and automating workflows. The paper also discusses best practices for using Boto3 effectively and highlights its role in enabling cloud-native development on the AWS platform.

Basic Concept

2.1 Cloud Computing:

The concept of cloud computing revolves around the provision of various computing services via the internet. This modern approach grants users the ability to utilize essential resources, including servers, storage solutions, databases, network capabilities, and software. The hallmark of cloud computing is its agility in provisioning and scaling services with minimal oversight, thus enabling organizations to concentrate on their primary operations. The spectrum of cloud computing encompasses models like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), each delineating a unique degree of control and managerial duties.

2.2 Database Management Systems (DBMS):

Database Management Systems (DBMS) are pivotal software tools that facilitate the creation, administration, and manipulation of databases. They serve as the interface through which users can engage with databases, performing actions such as query execution, data insertion, updates, deletions, and structural management. The realm of DBMS is diverse, with classifications such as relational, object-oriented, and NoSQL databases, each tailored to meet distinct requirements for data storage and retrieval.

2.3 Python:

Python stands out as an accessible, high-level programming language acclaimed for its straightforward syntax and legibility. It is adaptable to various programming paradigms, including procedural, object-oriented, and functional programming, which makes it a versatile tool for a broad array of applications. With a rich standard library and a plethora of third-party modules, Python is adept at handling tasks that span web development, data analytics, artificial intelligence, and scientific computing.

2.4 Flask:

Flask is a streamlined web framework for Python, designed to facilitate rapid web application development with a lean codebase. It equips developers with vital web development functionalities, including URL routing, request handling, and HTML templating. Adhering to the WSGI standard, Flask ensures compatibility across a variety of web servers and deployment environments.

2.5 Boto3:

Boto3 stands as the official AWS SDK for Python, offering a user-friendly interface for Python developers to engage with AWS services. It enables the automation of tasks such as cloud resource provisioning, storage management, and API calls. The extensive documentation and user-centric design of Boto3 have established it as a go-to tool for developing cloud-oriented applications and incorporating AWS services within Python-based projects.

2.6 MySQL:

MySQL is a renowned open-source RDBMS, extensively utilized for crafting scalable and resilient applications that are database-intensive. It boasts features like ACID compliance, transaction support, and indexing, catering to diverse scenarios ranging from modest web applications to comprehensive enterprise systems. The straightforward setup, management, and compatibility with prevalent programming languages render MySQL a favored option among developers.

2.7 HTML/CSS:

HTML and CSS constitute the core building blocks for crafting and styling web pages. HTML lays out the structural and content elements of web pages, delineating components such as headings, paragraphs, links, and images. In contrast, CSS orchestrates the visual and layout aspects, dictating color schemes, typography, and spatial arrangements. Collectively, HTML and CSS empower developers to create aesthetically pleasing and interactive web interfaces.

Chapter 3

Problem Statement / Requirement Specifications

The central challenge addressed by this project is the creation of a dynamic and scalable system for the effective management of student data within educational institutions. Current traditional practices in data management are often manual and prone to inefficiency, leading to inaccuracies and inconsistencies in student information. As educational institutions expand and become more complex, there is a pressing need for a unified, automated system to simplify administrative operations and ensure data precision.

3.1 Project Planning

- Outline the project's aims and boundaries.
- Determine the needs and expectations of all involved parties.
- Set forth a timeline and key deliverables for the project.
- Distribute tasks and define roles within the project team.

3.2 Project Analysis

With the project's requirements in hand and the problem statement defined, a detailed analysis is essential to uncover any uncertainties, contradictions, or potential obstacles. This stage is critical to confirm that the project's objectives are clear and meet the expectations of all stakeholders. This involves:

- Reassessing and fine-tuning the project requirements.
- Performing studies to assess the project's viability.
- Recognizing risks and formulating countermeasures.
- Confirming adherence to regulatory standards and ethical norms.

3.3 System Design

3.3.1 Design Constraints

The proposed system is designed to function within a cloud-based infrastructure, utilizing AWS services such as Amazon RDS for database management, Amazon S3 for data storage, and AWS Lambda for serverless operations. The development will be carried out using the Python programming language, alongside the Flask web framework for crafting the application. The system's hardware prerequisites are standard computing devices with reliable internet access.

3.3.2 System Architecture Overview

The architecture of the system is structured with several key components:

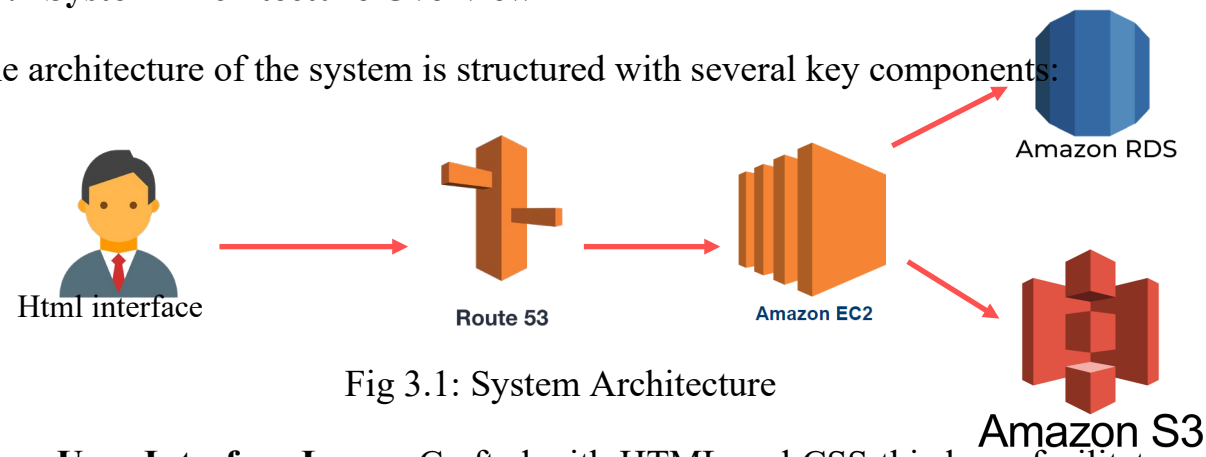

Fig 3.1: System Architecture

- **User Interface Layer**: Crafted with HTML and CSS this layer facilitates user engagement and interaction.
- **Application Logic Layer**: This layer, built with Flask, manages HTTP requests and orchestrates responses.
- **Data Management Layer**: Amazon RDS is employed here to securely house and administer student information.
- **Document Storage**: Amazon S3 is utilized for the secure storage of student-related documents and multimedia.
- **Serverless Operations**: AWS Lambda is harnessed to execute computational logic and data manipulation tasks.
- **Compute Layer:** EC2 instances are used for hosting the application and performing operations that require persistent compute capacity.
- **Service Integration**: Boto3, the AWS SDK tailored for Python, is integrated to ensure fluid interaction and operational harmony among AWS services.

Chapter 4

4.1 Methodology OR Proposal

The approach for executing the project was methodically structured and encompassed the following phases:

- **Requirement Analysis**: A thorough examination of the requirements of educational bodies and their stakeholders was conducted to delineate the project's aims and boundaries.
- **Design Phase**: The development of a coherent system architecture and a structured database schema was undertaken to efficiently organize and manage student data.
- **Development Stage**: The coding process involved crafting both the frontend and backend features, utilizing technologies such as Flask, HTML, CSS and Python.
- **Integration Process**: A seamless fusion of AWS services, including Amazon RDS, Amazon S3, and EC2 instances, was achieved to facilitate smooth system operations and communication.
- **Testing Protocol**: A series of stringent tests employing specific test cases were applied to ascertain the system's integrity and performance.
- **Deployment Procedure**: The final step involved deploying the application on an EC2 instance, thereby making it available for internet access.

4.2 Testing OR Verification Plan

The testing phase involved the creation and execution of various test cases to validate the system's functionality. A sample test case table is provided below:

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Add Student	Valid Input	Student added	Success
T02	Fetch Student	Existing ID	Student fetched	Success
T03	Update Student	Valid data	Data updated	Success

4.3 Result Analysis OR Screenshots

This section presents a visual and analytical representation of our project's outcomes. It includes captured images of the application's interface, such as the form for adding student information, the detailed student profile page, and the structured database tables. These visuals serve to illustrate the practicality and user experience offered by the system.

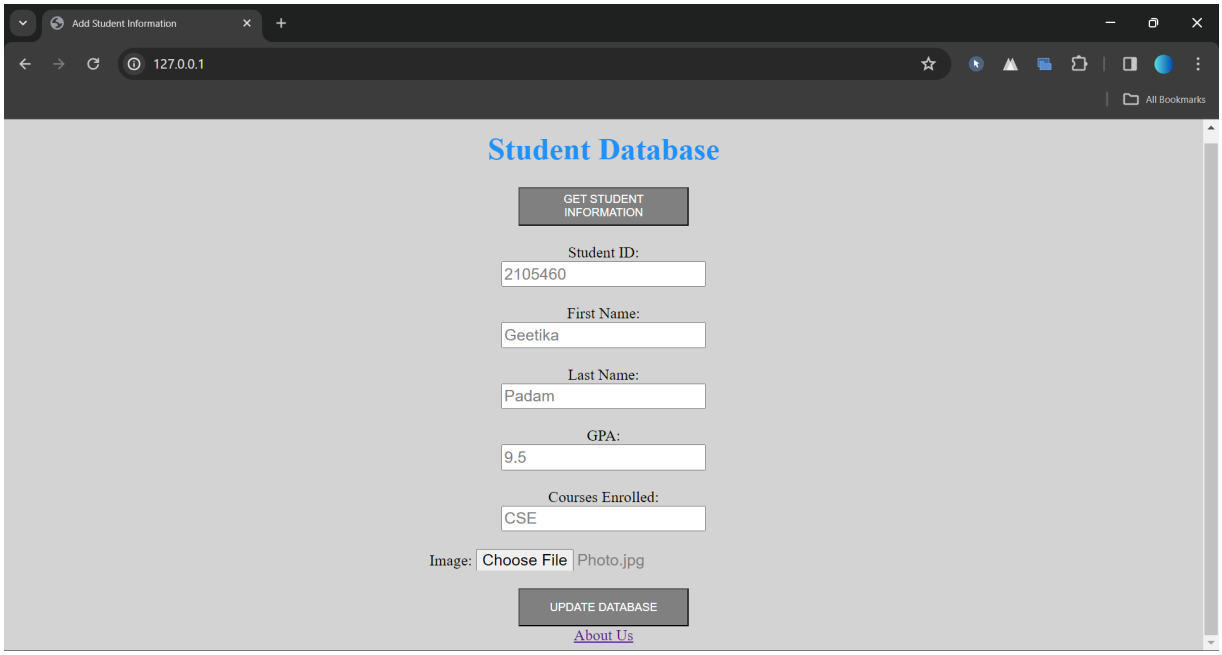

Fig 4.3.1: User Interface to Add Data

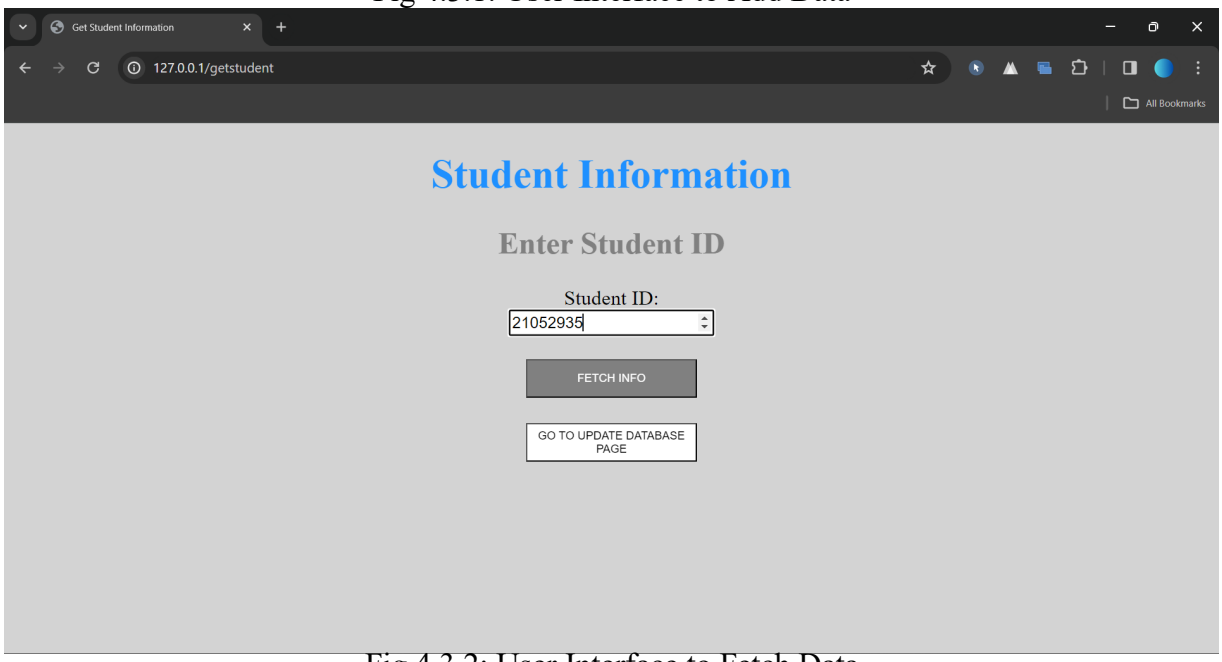

Fig 4.3.2: User Interface to Fetch Data

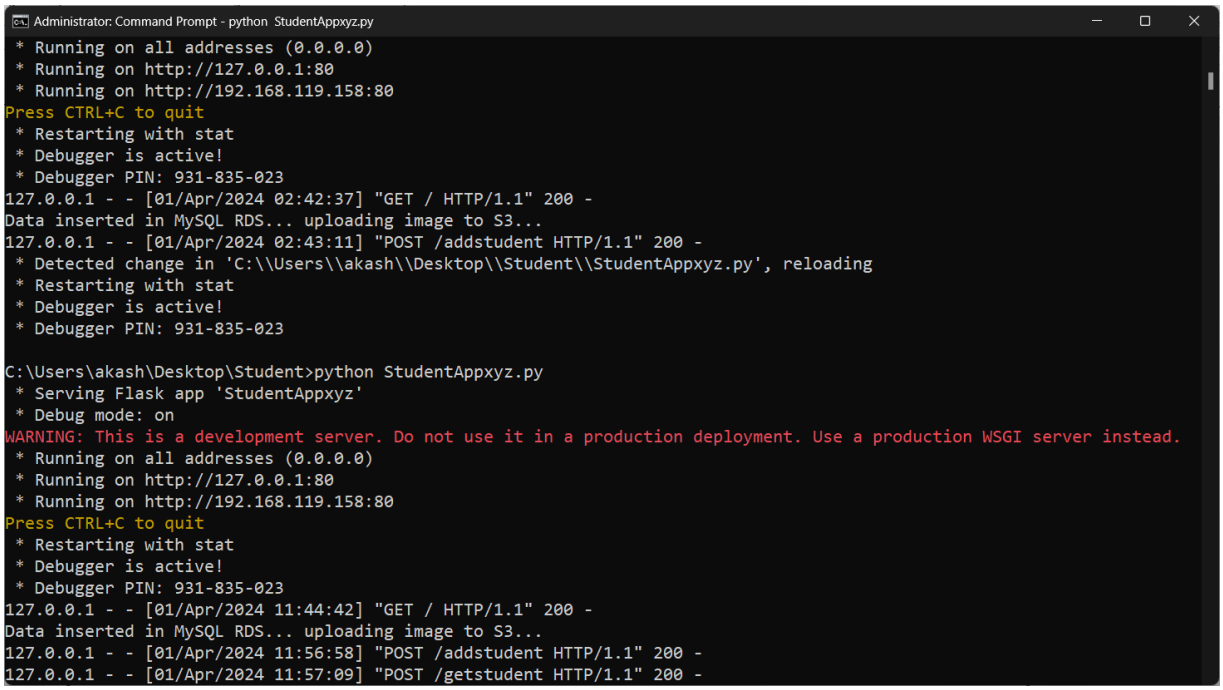

Fig 4.3.3: Connection from Local Host terminal

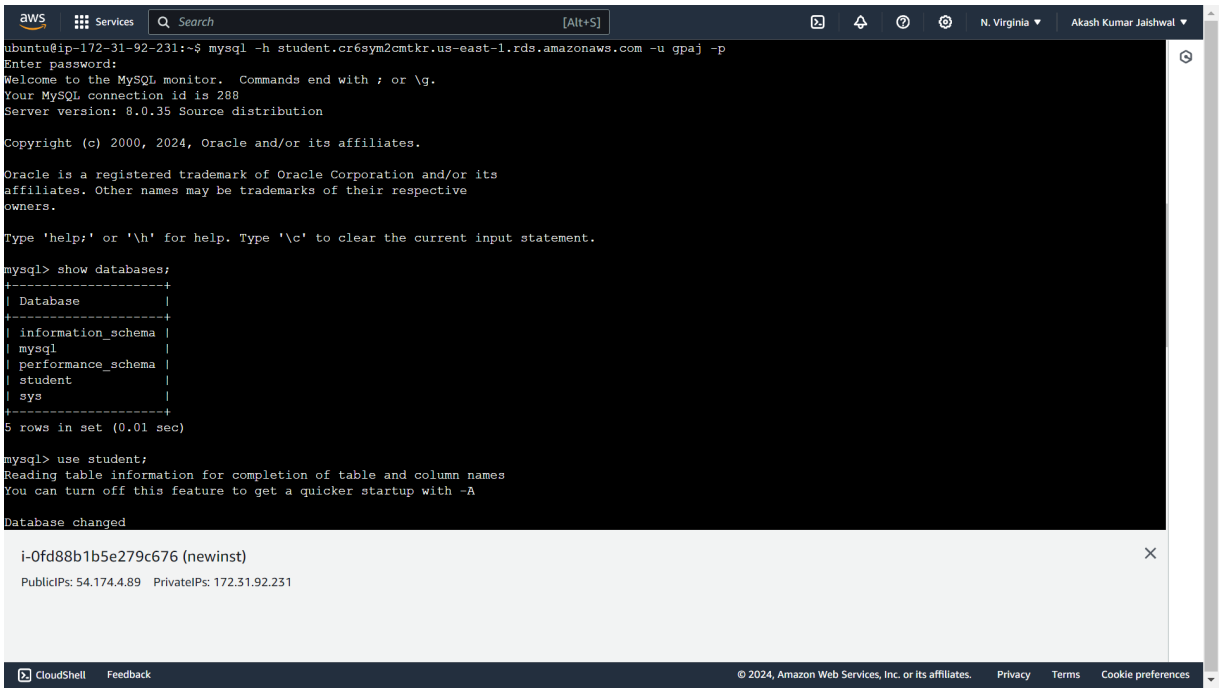

Fig 4.3.4: EC2 instace Run and Connection with SQL Database

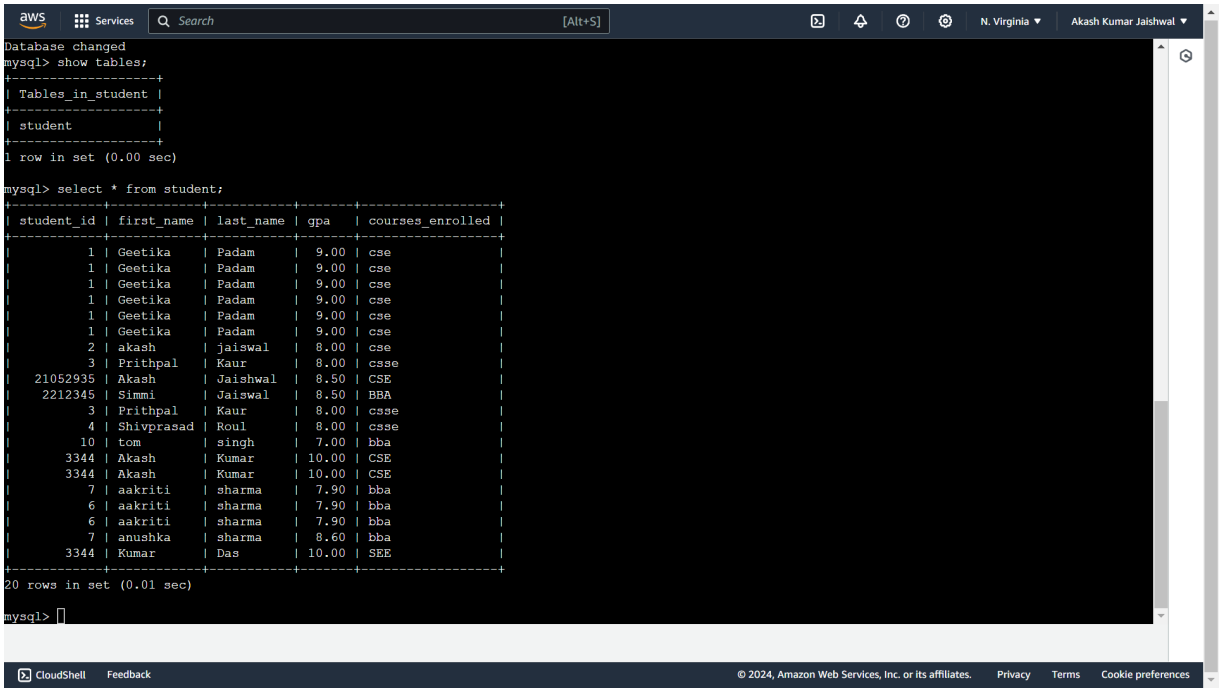

Fig 4.3.5: Database Table display after data Input

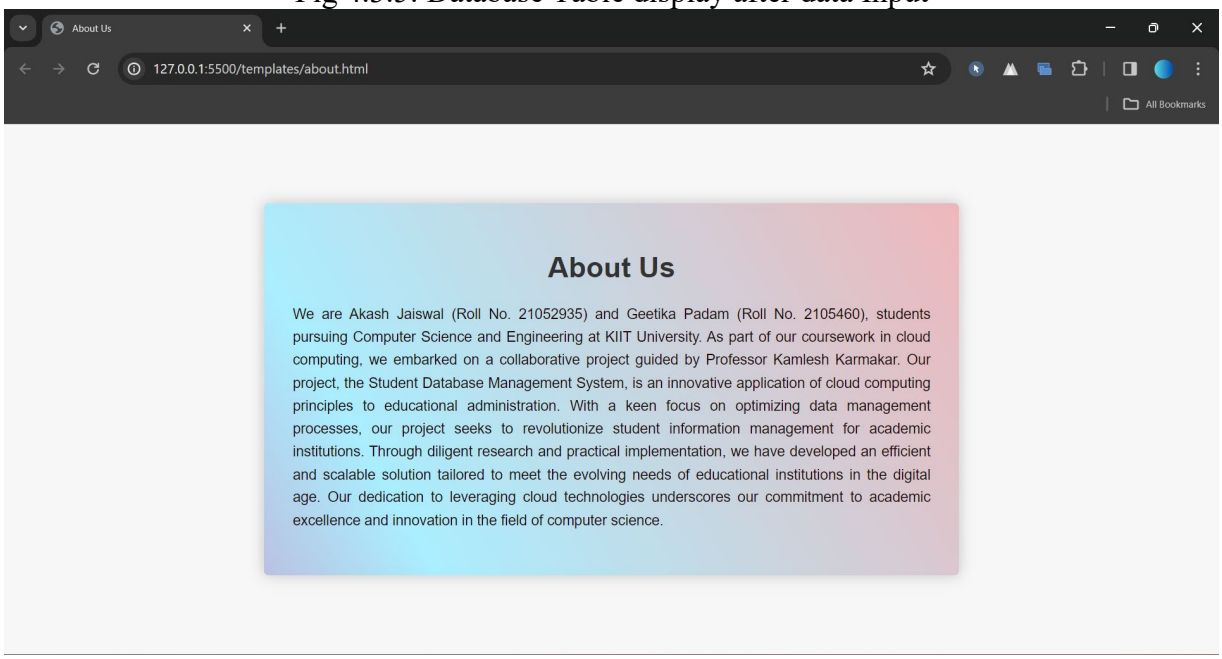

Fig 4.3.6: About Us

4.4 Quality Assurance

The project's quality assurance strategy encompassed a series of systematic code evaluations, comprehensive testing, and thorough validation checks. This rigorous process was essential to confirm that the project adhered to its predefined objectives and quality benchmarks. Furthermore, the development process was consistently aligned with industry best practices and established coding standards to maintain high-quality deliverables throughout the project's progression.

Chapter 5

Standards Adopted

5.1 Design Protocols

Throughout the design stage of our project, we committed to a framework of recognized protocols and best practices, which included:

- **IEEE Guidelines:** Our application's architectural design was crafted in accordance with IEEE guidelines, ensuring compliance with globally acknowledged standards of quality and dependability.
- **UML Visualization:** We employed Unified Modeling Language (UML) tools, such as use case, class, and sequence diagrams, to effectively delineate and convey our application's design.

5.2 Programming Guidelines

Our coding approach was guided by a series of principles aimed at preserving the integrity and clarity of our code:

1. **Clarity:** Our objective was to maintain clear and succinct code, steering clear of unnecessary intricacies.
2. **Standardized Naming:** We implemented standardized naming conventions for variables, functions, and classes to enhance the readability and upkeep of our code.
3. **Structured Code:** We organized our code into well-defined segments, utilizing proper indentation to foster a clear code hierarchy.
4. **Componentization:** We adopted a modular approach, partitioning our code into smaller, interchangeable units to facilitate reuse and maintenance.
5. **Optimized Functions:** We focused on creating functions with a targeted scope to aid in the maintainability and comprehension of our code.

5.3 Verification Protocols

For the testing phase, we embraced established protocols and methodologies to ascertain the robustness and dependability of our application:

- **ISO Compliance:** We conformed to ISO testing standards, aligning our quality assurance processes with international benchmarks.
- **IEEE Testing Guidelines:** We also adhered to IEEE testing standards, applying meticulous testing techniques to ensure the application's functionality and efficiency.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

To summarize, our initiative, the Student Database Management System, has adeptly resolved the core issue by harnessing the power of cloud computing to refine the management of student data. Our team's extensive research, careful planning, and steadfast implementation have given rise to a potent and scalable system that satisfies the data management demands of scholastic institutions. Incorporating Amazon RDS for robust data storage, Amazon S3 for file handling, and Amazon EC2 for agile web application hosting, we've forged an all-encompassing platform that bolsters data access, fortifies security, and ensures steadfast reliability. Moreover, this endeavor accentuates the pivotal role of cloud computing in revolutionizing educational management and demonstrates the myriad advantages that cloud-based technologies offer in boosting operational efficacy and scholarly productivity.

6.2 Future Scop

Prospective Developments: Our venture has reached its initial targets, yet the horizon beckons with opportunities for further advancements and extensions:

Robust Security Enhancements: We aim to fortify our system's defenses by deploying sophisticated encryption methods and layered verification processes, ensuring the integrity and confidentiality of student data.

Analytical Integration: By embedding advanced analytics into our platform, we can mine educational data for insights, uncovering trends and informing strategic decisions within academic institutions.

Mobile Platform Extension: The creation of a mobile extension for our Student Database Management System will enable educators, students, and administrators to access vital information anytime, anywhere.

LMS Connectivity: Fusing our solution with leading Learning Management Systems will streamline the management of educational content and student databases, offering a unified user experience.

AI-Driven Predictive Models: Utilizing artificial intelligence to anticipate academic trajectories, we can identify students needing support early on and tailor interventions to bolster their academic journey.

Embracing these paths for future growth will not only elevate the capabilities and user experience of our Student Database Management System but also significantly contribute to the progressive transformation of educational administration and the enhancement of student achievement initiatives.

References

1. Amazon Web Services. (n.d.). AWS Documentation.
Retrieved from https://docs.aws.amazon.com/

2. Python Software Foundation. (n.d.). Python Documentation.
Retrieved from https://docs.python.org/

3. Flask Documentation. (n.d.). Flask Documentation.
Retrieved from https://flask.palletsprojects.com/en/2.0.x/

4. MySQL. (n.d.). MySQL Documentation.
Retrieved from https://dev.mysql.com/doc/

5. W3Schools. (n.d.). HTML Tutorial.
Retrieved from https://www.w3schools.com/html/