Microsoft Elevate

CAPSTONE PROJECT

Cloud-Based Student Performance Management System using Microsoft Azure

PRESENTED BY : - DONTHU HARSHAVARDHAN

STUDENT NAME :- DONTHU HARSHAVARDHAN

COLLEGE NAME :- BITS – (JNTUA) - KURNOOL

DEPARTMENT :- CSE

EMAIL ID: DONTHUHARSH2004@GMAIL.COM

Microsoft Elevate

OUTLINE:

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

Microsoft Elevate

PROBLEM STATEMENT:

- Educational institutions often face difficulty in managing, analyzing, and accessing student academic performance data efficiently.

  Traditional systems are manual, error-prone, and lack real-time access, scalability, and data security.

- There is a need for a **cloud-based, centralized system** to store, retrieve, and analyze student performance data securely and efficiently

Microsoft Elevate

PROPOSED SOLUTION:

- The proposed system is a **cloud-based Student Performance Management System** built using **Microsoft Azure services**.

  It provides a centralized platform to store, manage, and analyze student marks securely.

- The system enables:
- Secure storage of student academic records in **Azure SQL Database**
- RESTful API developed using **Python Flask**
- Data access and management through a **web-based dashboard**
- Cloud deployment using **Azure App Service**
- Scalability, reliability, and real-time accessibility
- This solution reduces manual effort and improves data accuracy and availability.

Microsoft Elevate

SYSTEM APPROACH:

- The project follows a **modular and cloud-first development approach**.
- **Steps involved:**
- **Requirement Analysis**
  - Identify data fields (StudentID, Name, Semester, Subject, Marks, TotalMarks)
  - Define functional requirements (CRUD operations, filtering, dashboard access)
- **Database Design**
  - Design relational schema using **Azure SQL Database**
  - Create StudentMarks table with appropriate data types
- **Backend Development**
  - Develop REST APIs using **Python Flask**
  - Implement CRUD operations (Add, View, Update, Delete)
  - Connect Flask to Azure SQL using **pyodbc**
- **Frontend Development**
  - Build simple HTML templates (Login, Dashboard, Add/Edit Student)
  - Integrate frontend with backend APIs
- **Cloud Integration**
  - Deploy application on **Azure App Service**
  - Configure environment variables securely
- **CI/CD & Deployment**
  - Source code hosted on **GitHub**
  - Automated deployment using **GitHub Actions**

ALGORITHM & DEPLOYMENT:

- **Algorithm**
- The system uses a **rule-based data processing approach** (no ML model).
- **Algorithm Flow:**
- User sends a request (login / add / update / delete / view records)
- Flask API validates input data
- SQL queries are executed on **Azure SQL Database**
- Retrieved data is processed and formatted
- Response is sent back to the frontend/dashboard
- **Key Operations:**
- INSERT → Add student marks
- SELECT → Fetch student records & filters
- UPDATE → Modify existing records
- DELETE → Remove records

Microsoft Elevate

Deployment

- Backend deployed on **Azure App Service**
- Database hosted on **Azure SQL Database**
- Secure connection via environment variables
- CI/CD implemented using **GitHub Actions**
- Application accessible via public Azure URL

Microsoft Elevate

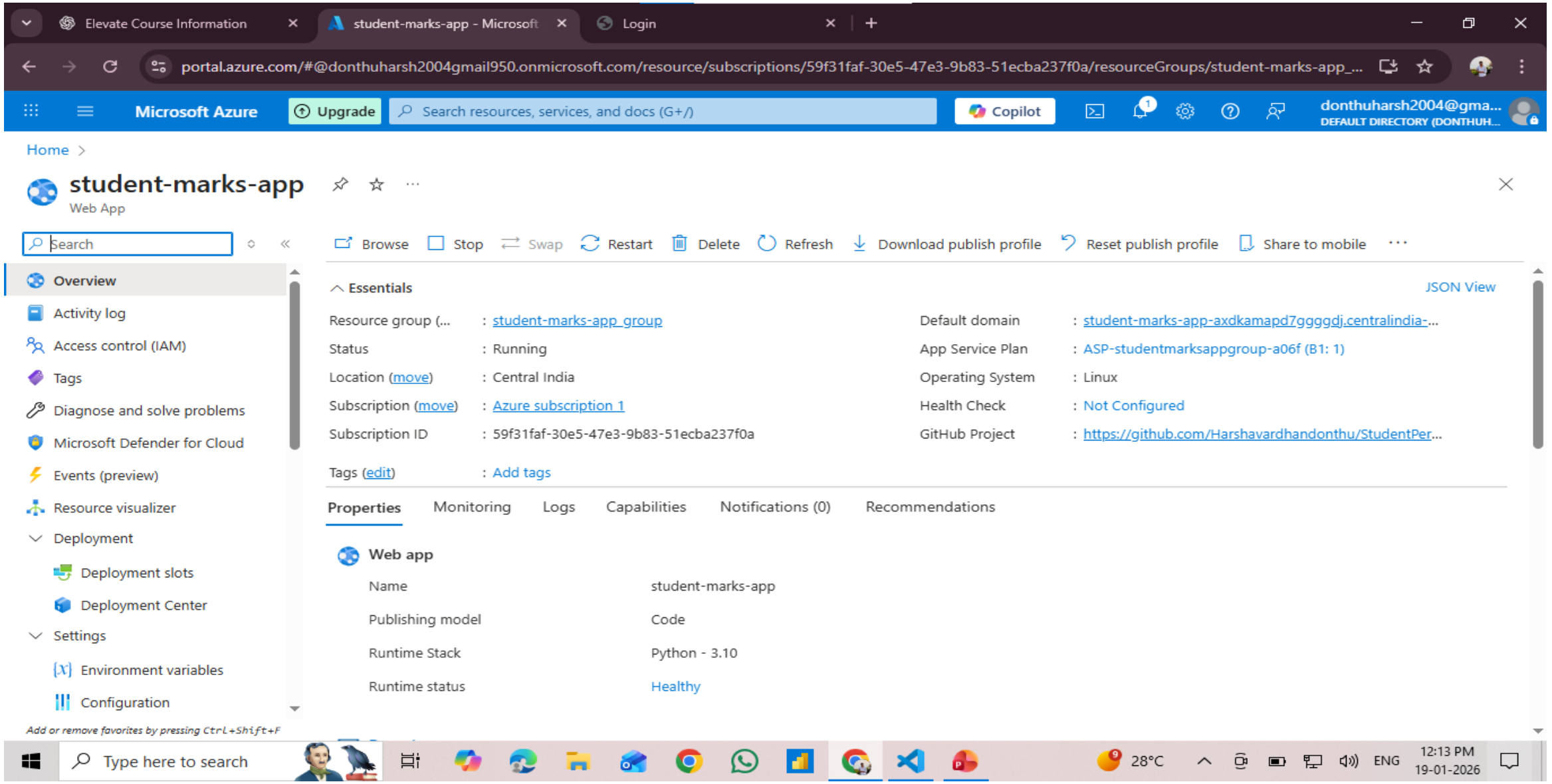
RESULT:

- Successfully developed and deployed a **cloud-based student performance system**
- Student data is stored securely in **Azure SQL Database**
- Users can **add, view, update, and delete** student marks through the web interface
- REST APIs respond in real time with accurate data
- Application is accessible globally using an **Azure-hosted URL**
- GitHub-based deployment ensures smooth updates and version control

The system works reliably with proper authentication and database connectivity.

Git Hub repo link:https://github.com/Harshavardhandonthu/StudentPerformanceAPI.git

Microsoft Elevate



Microsoft Elevate

CONCLUSION:

- The Cloud-Based Student Performance Management System successfully demonstrates the use of **Microsoft Azure** for building and deploying a scalable web application.
  By integrating **Flask**, **Azure SQL Database**, and **Azure App Service**, the system provides secure and efficient management of student academic data.
- The project highlights practical knowledge of:
- Cloud computing concepts
- Backend development
- Database management
- CI/CD deployment using GitHub
- Overall, the solution meets the project objectives and provides a reliable cloud-based data management platform.

Microsoft Elevate

FUTURE SCOPE:

- The system can be further enhanced with the following features:
- Role-based authentication for **Admin, Faculty, and Students**
- Advanced analytics and visualization using **Power BI**
- Export reports in **PDF or Excel** format
- Integration of **machine learning** for performance prediction
- Mobile-friendly user interface
- Deployment using **Docker and Azure Container Services**
- These enhancements will improve scalability, usability, and analytical capabilities.

Microsoft Elevate

REFERENCES:

- Microsoft Azure Documentation
  https://learn.microsoft.com/azure
- Flask Official Documentation
  https://flask.palletsprojects.com
- Azure SQL Database Documentation
  https://learn.microsoft.com/azure/azure-sql
- GitHub Actions Documentation
  https://docs.github.com/actions
- Python Official Documentation
  https://www.python.org/doc/

Microsoft Elevate

Thank You