Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

CE308 – CLOUD COMPUTING

Project Title:

Cloud-Based Campus Complaint Management System

Abdul Mueed Khan
2022013
B.Sc. Computer Science
Ghulam Ishaq Khan Institute
Topi, Swabi
u2022013@giki.edu.pk

Muazzam Shah
2022312
B.Sc. Computer Science
Ghulam Ishaq Khan Institute
Topi, Swabi
u2022312@giki.edu.pk

Table of Contents

Abstract

This project introduces a cloud-hosted Hostel Complaint Management System developed to streamline and digitize the process of reporting and managing hostel-related issues within the GIKI campus. Utilizing modern web technologies and scalable cloud infrastructure, the system provides a seamless platform where hostel residents can submit complaints, track progress, and view resolution status in real-time. The backend is powered by a Flask application deployed using AWS Elastic Beanstalk, and the data is managed through a PostgreSQL database hosted on AWS RDS. This project demonstrates the integration of cloud-based services to achieve reliability, transparency, and administrative efficiency.

Introduction

In campus environments like GIKI, frequent issues such as plumbing failures, electrical problems, and maintenance delays often impact the quality of life for hostel residents. Traditional complaint mechanisms—such as handwritten logs or verbal reports—lack traceability and accountability, resulting in inefficient resolutions.

To address these problems, we have developed a Cloud-Based Hostel Complaint Management System that leverages the scalability and reliability of cloud services. The system allows students to submit complaints digitally, while administrators can assign, monitor, and resolve issues through a centralized dashboard. It ensures transparency and promotes faster issue resolution by tracking the status of each complaint from submission to closure.

Project Objectives

- **Dedicated Platform for Hostel Complaints:** A student-friendly portal to report issues related to hostel facilities.

- **Efficient Complaint Routing:** Each complaint is categorized and made visible to admin staff.

- **Real-Time Complaint Tracking:** Students can monitor the progress of their submissions.

- **Transparency and Accountability:** Detailed logs and complaint history improve trust and administrative clarity.

- **Scalable and Reliable Infrastructure:** Built on cloud infrastructure to support multiple hostels and concurrent users.

System Features

Student (User) Side

- **Complaint Submission:** Users can log complaints by selecting a category (e.g., Plumbing, Electricity) and describing the issue.

- **Status Tracking:** View current status (Pending, Assigned, In Progress, Resolved).

- **Complaint History:** Access a log of previously submitted complaints.

Admin/Management Side

- **Complaint Dashboard:** Admins view all complaints and filter by hostel, category, or status.

- **Status Update:** Change status with remarks for traceability.

System Workflow

1. **User Authentication:**
   Users log in through Firebase Authentication.

2. **Complaint Submission:**
   Student submits a complaint via a form selecting a category and description, optionally uploading an image.

3. **Data Storage:**
   Complaint data is saved in AWS RDS, and image (if uploaded) is stored in S3.

4. **Admin Dashboard Access:**
   Admins access a dashboard to view, assign, and update complaints.

5. **Complaint Closure:**
   Once resolved, the complaint is marked as "Resolved" and archived.

Cloud Deployment Details

The **Hostel Complaint Management System** was deployed using multiple cloud services provided by AWS and Firebase. This deployment ensured that the application is **scalable, secure, and highly available**. Below are the detailed components of the deployment:

1. Backend Deployment – AWS Elastic Beanstalk

The backend is developed using **Flask**, a Python micro-framework, and is deployed using **AWS Elastic Beanstalk**, which abstracts server management and offers auto-scaling, load balancing, and health monitoring.

Key Aspects:

- **Environment Setup:**

  o A Python 3.11 environment was configured on Elastic Beanstalk.

  o The application was packaged with the required dependencies using requirements.txt.

  o A Procfile was included to define the startup command (*web: gunicorn application:app*).

- **Environment Variables:**

  o Critical configuration values like the PostgreSQL database URI, AWS S3 credentials, Firebase keys, and Flask secret key were **not hardcoded**.

  o These were instead securely injected using the **Elastic Beanstalk environment configuration console**.

- o Example variables set:
    - DATABASE_URL
    - AWS_ACCESS_KEY_ID
    - AWS_SECRET_ACCESS_KEY
    - S3_BUCKET_NAME
    - FIREBASE_API_KEY
    - FLASK_SECRET_KEY

These were accessed inside the flask application. This practice keeps credentials secure and makes the app cloud-portable.

2. Database – AWS RDS (PostgreSQL)

The application uses **Amazon RDS with PostgreSQL** to store user complaints, complaint statuses, and user data.

Configuration:

- **Instance Class:** db.t3.micro (sufficient for project scale).
- **Storage:** Allocated up to 20GB with automatic backups enabled.
- **Security:**
  - o Database is **not publicly accessible**.
  - o A security group rule was added to allow inbound traffic **only from Elastic Beanstalk instances**.

3. Authentication – Firebase Authentication

User authentication (login/register) is implemented using **Firebase Authentication**, which provides secure and scalable identity services.

Features:

- **Email/Password Authentication** for both students and admins.
- **Token Verification:** JWT tokens issued by Firebase are verified in the backend to confirm the user's identity before allowing sensitive actions (e.g., submitting or viewing complaints).

4. File Storage – AWS S3 Bucket

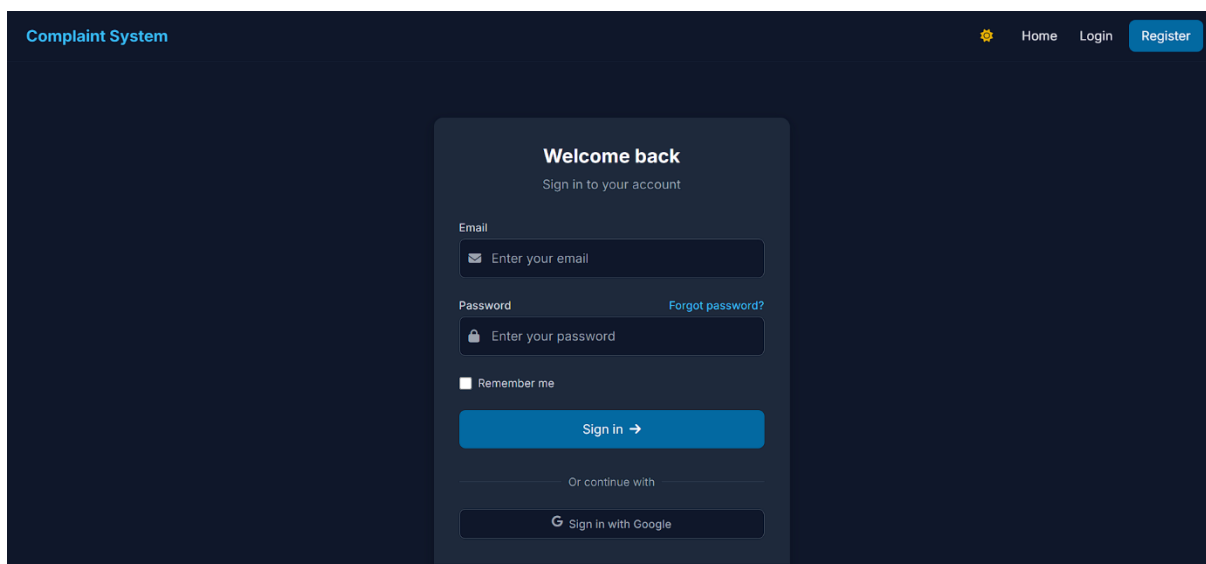Users can optionally upload images along with complaints.

Configuration:

- An **AWS S3 bucket** was created with:

- o **Bucket policy** allowing access only through signed URLs or authenticated API calls.

- o **Public access** was disabled to ensure image privacy.

- **Integration:**

  - o Uploaded files are sent to the backend and stored in S3.

- **Access Control:**
  - o AWS credentials are never hardcoded.
  - o IAM policies assigned to the Beanstalk EC2 instance through its role allow **write-only** access to the S3 bucket, ensuring the app cannot list or delete files arbitrarily.
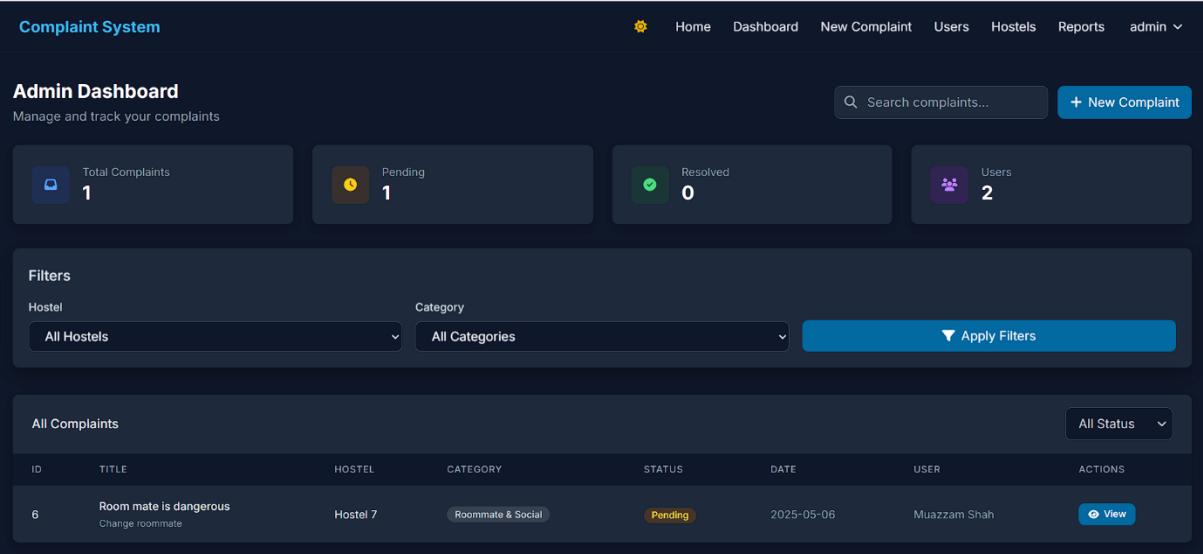
Screenshots and UI Overview

Login Page:

Dashboard:

Complaint:

Status:

Source Code

Github Link:

https://github.com/Muazzam-Shah/Complaint-Management-System

Deployed App:

http://giki-complaint-env.eba-v4cqjnsg.eu-north-1.elasticbeanstalk.com/

Conclusion

The Cloud-Based Hostel Complaint Management System successfully addresses the inefficiencies and delays associated with traditional manual complaint handling at GIKI. By leveraging a cloud-native architecture, the system provides a centralized, reliable, and scalable platform where hostel residents can submit complaints and track their progress in real time.

The use of **AWS services** such as **Elastic Beanstalk**, **RDS**, and **S3**, combined with **Firebase Authentication**, allowed us to build a secure, modular, and cloud-optimized system. Key benefits include seamless access across devices, enhanced transparency between students and administration, and the ability to manage complaint records efficiently with minimal downtime.