MAKERERE UNIVERSITY

MUSIIME DORIS	21/U/06444/EVE
AINOMUGISHA PRISCILLA	21/U/17675/EVE
WAFULA IAN ELMER	21/U/07949/EVE
SSERUNKUMA CHARLES ROLAND	21/U/05667/EVE
ALEMI BLESS	21/X/20013/EVE

Table of Contents

Introduction.

The Student Feedback Django Web Application is a comprehensive tool designed to facilitate the collection, analysis, and reporting of feedback from students about various aspects of an educational institution. This documentation provides an in-depth overview of the web application, its architecture, features, installation steps, and usage guide.

System Overview.

Assumptions:

- The student is already registered with the University and uses his/her login credentials(student number and password) to log into the system.
- The courses and course units are from the university database
- The administrators are already registered with the university hence use there credentials to log into the system

Architecture.

The web application follows the model-view-template (MVT) architecture pattern, with a strong emphasis on the Django framework's built-in components.

- Model: represents the data structure and interacts with the database.
- View: defines how data is presented to users.
- Template: This utilises django's templating system

Technologies

Django framework: Provides the foundation for building the web application.
Python: The primary programming language used for server-side logic.
HTML, CSS, JavaScript: Used for front-end development and user interface.
MySQL: The chosen database management system for storing feedback and log in data.
Streamlit: Utilized for data visualization and analytics.

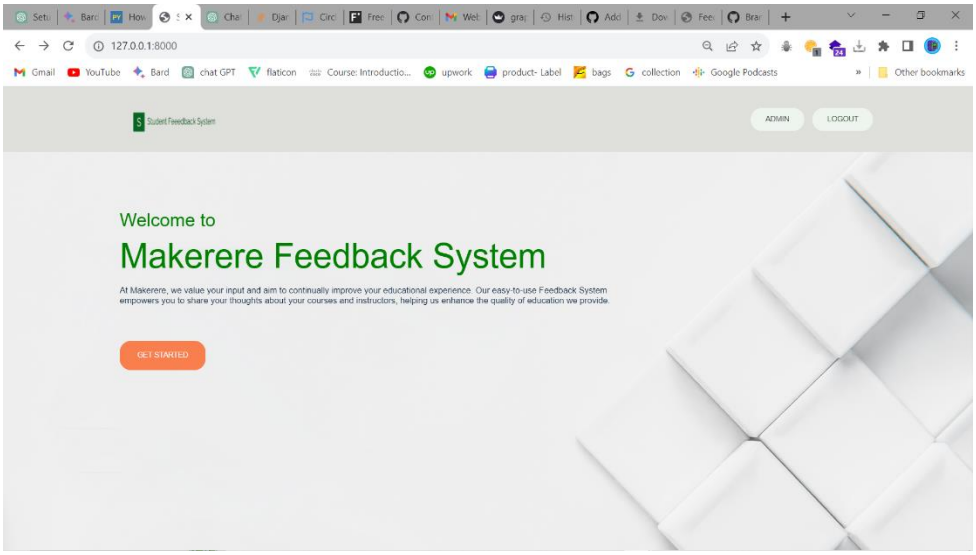
Features.

User roles.

- **Admin**: The admin visualises and analyses the feedback from the feedbacks and makes sense of it.
- **Student**: Only provides feedback.

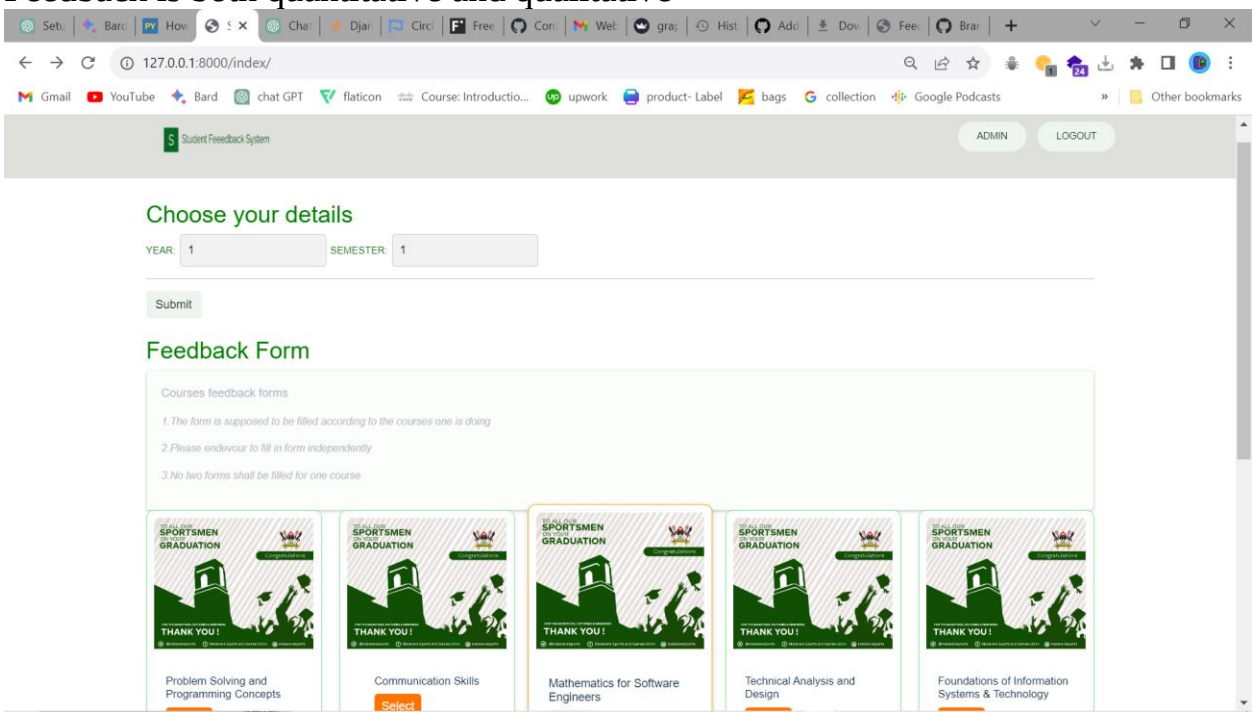
Landing Page:

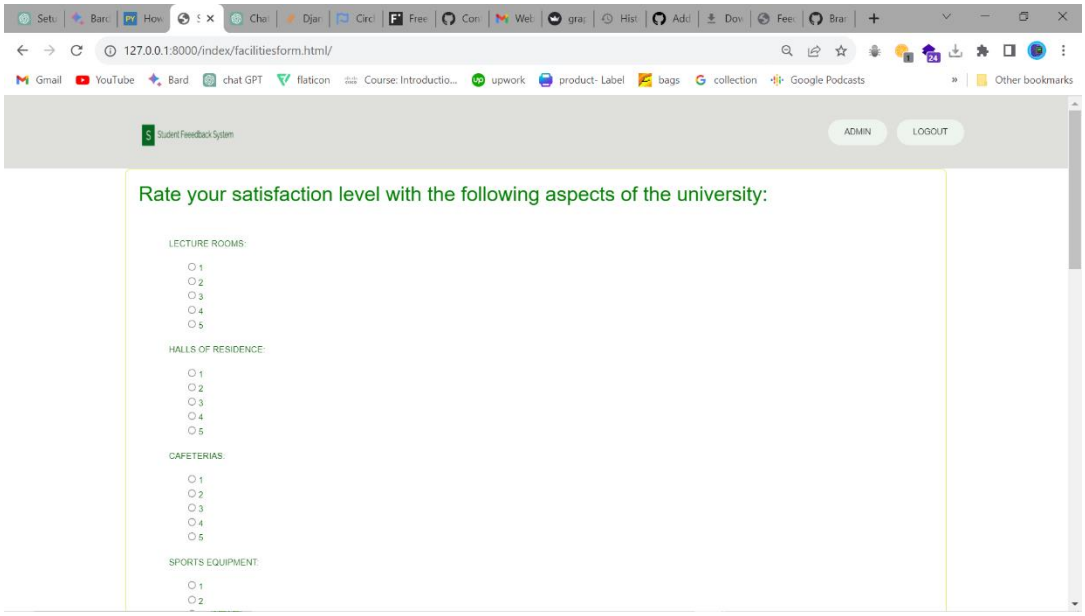
- This is where users are redirected to When they come to the site



Feedback submission.

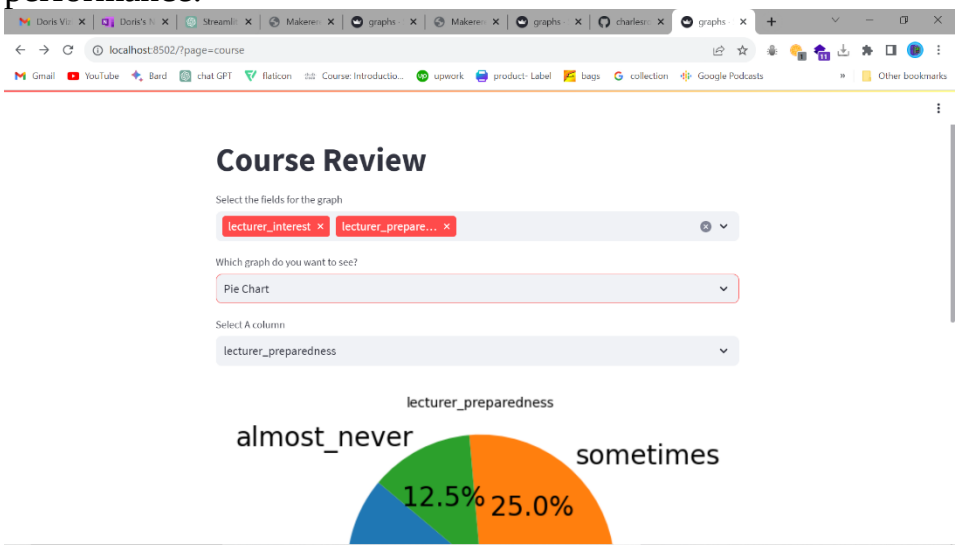
- Students can provide feedback on courses, instructors and campus facilities.
- Feedback is both quantitative and qualitative

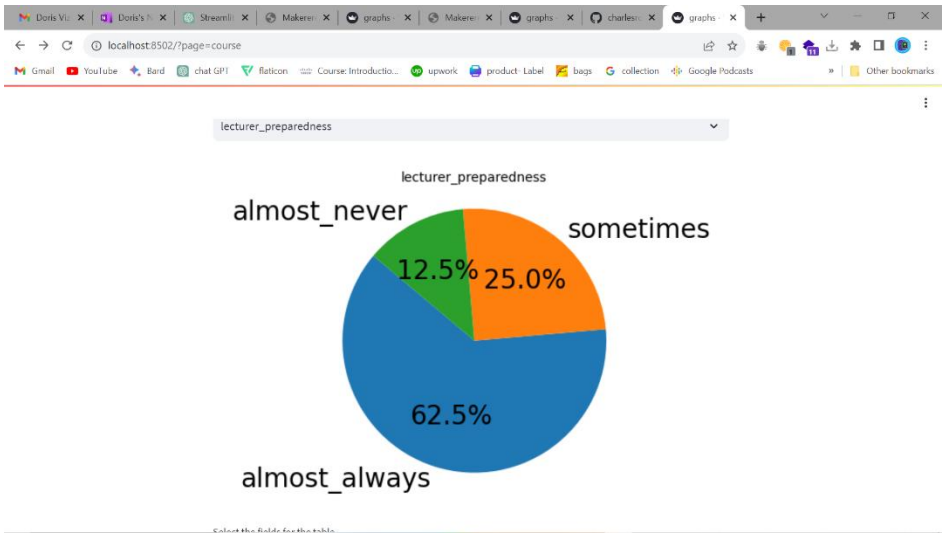




Data Management.

- All feedback submissions are stored in a MySQL database under there respective tables.
- Data integrity and security are maintained through Django's built-in mechanisms.
- Analytics and reporting.
- Admin users can generate reports and visualize trends using charts.
- Reports provide insights into course and facility effectiveness, and instructor performance.





Installation and setup.

Prerequisites:

- Django
- Streamlit
- Bootstrap
- Crispy-forms
- Mysql
- python

Usage guide.

- User login.
- Access the application via a web browser.
- Register as a student or admin using the provided forms.
- Log in using your credentials.

- Providing feedback.
- Students can submit feedback by selecting a year and semester.
- The system then displays the course units that match the year and semester.
- There are two feedback forms, one for courses and instructors and the other for facilities.
- The feedback is then submitted and saved in a database

- Viewing feedback.

- Admin view gives an overview of the system, managing the visualizations and analysis.
- Charts provide visual insights into feedback trends.
- The table can be queried for specific data.
- Reports can be filtered by date, course or year.

Troubleshooting.

- Ensure all prerequisites are properly installed.
- Check database configurations in 'settings.py'.
- Review error messages for guidance.

Future enhancements.

- Integration with email notifications for admins.
- Possibility of adding api's to the system for other end users
- Multi-language support for international users.
- Integration with external tools for enhanced analytics.

Conclusion.

The student feedback Django we application serves as a valuable tool for educational institutions to gather and analyze feedback from students, facilitating continuous improvement in course quality and instructor performance.

References.

Django documentation: https://docs.djangoproject.com/
MySQL Documentation: https://dev.mysql.com/
Streamlit Documentation: https://docs.streamlit.io/