

Project

to Design and Implementation

of a Student Management System for Teachers

by

Shri Ram Dwivedi

Bachelor of Technology in Computer Science with Artificial Intelligence & Machine Learning

3rd Semester 2024-25

Roll No. 24123565

Submitted to

Computer Science & IT Department

Kalinga University

Naya Raipur

Guided by

Mrs. Roohee Khan

Assistant Professor, Kalinga University

**DECLARATION BY THE CANDIDATE**

I the undersigned solemnly declare the report of the thesis work entitled “Design and Implementation of a Student Management System for Teachers” is based on my own work carried out during the course of my study under the supervision of Mrs. Roohee Khan.

I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/deemed University of India or any other country. All helps received and citations used for the preparation of the thesis have been duly acknowledged.

(**Signature of Candidate)**

Shri Ram Dwivedi

Roll. 24123565

Enrolment No. 23071350985983

**CERTIFICATE OF THE SUPERVISOR**

This is to certify that the report of the thesis entitled Design and Implementation of a Student Management System for Teachers is a record of bonafide research work carried out by Shri Ram Dwivedi bearing Roll No.:24123565 Enrolment No.: 23071350985983 under my guidance and supervision for the award of Degree of Bachelor of Technology of Kalinga University, Atal Nagar (C.G.), India

* To the best of my knowledge and belief the thesis
* Embodies the work of the candidate himself,
* Has duly been completed,

Fulfils the requirement of the Ordinance relating to the Btech degree of the University and is up to the desired standard both in respect of contents and language for being referred to the examiners.

**(Signature of Head of Department)** **(Signature of Supervisor)**

Dr. O.P. Dewangan Mrs. Roohee Khan

(Head of Department) (Assistant Professor)

Computer Science & IT Department

Kalinga University

Naya Raipur

**CERTIFICATE BY THE EXAMINERS**

The Thesis entitled Design and Implementation of a Student Management System for Teachers project Submitted by Shri Ram Dwivedi bearing Roll No.24123565 and Enrollment No.: 23071350985983 has been examined by the undersigned as a part of the examination and is hereby recommended for the award of the degree of Bachelor of Technology in the faculty of CS & IT of Kalinga University, Atal Nagar (C.G.)

Internal Examiner External

Date: Date:

**ACKNOWLEDGEMENT**

I express my sincere gratitude to many people who have helped me and supported during project work. Without them I could not have completed the project on time. I am thankful to my guide and the of CS & IT, Mrs. Roohee Khan for valuable guidance, encouragement and patience.

I thank all my colleagues and friends for their cooperation while completing this project work. I want to thank my family members, without whose emotional and moral support nothing was possible.

Computer Science & IT Department

Kalinga University

Naya Raipur

**Abstract**

The rapid growth of educational institutions necessitates efficient management systems for handling vast amounts of student data. This project focuses on the design and implementation of a Student Management System tailored for teachers, providing a user-friendly interface to manage student information, including academic results. Leveraging Python's Tkinter for the graphical user interface (GUI) and MySQL for robust data storage, this system aims to streamline administrative tasks, ensuring reliability and ease of use.

The main objectives of the project include the development of functionalities for adding, updating, and viewing student records, a secure login system for teachers, and intuitive navigation within the application. The project is designed to be scalable, allowing for future enhancements and integration of additional features.

This thesis covers the comprehensive development process, from system architecture and database design to implementation and testing. The testing phase demonstrates the system's reliability and efficiency in handling student data. The challenges faced during development and the solutions implemented are discussed in detail, providing insights into the problem-solving approaches.

The result is a robust, user-friendly Student Management System that enhances the efficiency of managing student data for teachers, contributing significantly to the administrative processes in educational institutions. Future work will focus on expanding the system's capabilities, incorporating additional features, and improving user experience based on feedback.

**Table of Content**

|  |  |  |
| --- | --- | --- |
| S.NO. | Content | Page No. |
| 1 | Title Page | 1 |
| 3 | Declaration by the Candidate | 2 |
| 4 | Certificate of the Supervisor | 3 |
| 5 | Certificate by the Examiners | 4 |
| 6 | Acknowledgement | 5 |
| 7 | Abstract | 6 |
| 8 | Table of Contents | 7 |
| 9 | Introduction   * Background * Objectives * Scope | 8 |
| 10 | System Design   * Architecture * Components * Data Flow * Visual Representation of Database * Database Design * User Interface | 9 |
| 11 | Implementation   * Tools and Technologies * Modules * User Interface * Implementation Steps | 11 |
| 12 | Testing   * Test Cases * Results and Analysis | 14 |
| 13 | Conclusion   * Summary | 16 |
| 14 | References | 17 |
| 15 | Appendices   * Source Code | 18 |

**Introduction**

**Background**

Managing student data is essential for schools. Traditional methods can be slow and error-prone. A digital system can help make this process easier and more accurate. This project aims to create a user-friendly tool for teachers to manage student information effectively.

**Objectives**

The main goals of this project are:

* Easy Interface: Build a simple and intuitive interface for teachers using Python's Tkinter.
* Secure Storage: Use MySQL to safely store student data.
* Manage Records: Allow teachers to add, update, view, and delete student records.
* Secure Access: Ensure only authorized teachers can access the data.

**Scope**

The project focuses on the core features needed to manage student records. It is designed to be scalable, so more features can be added in the future. The key functions include:

* Secure login for teachers.
* Managing student records.
* Viewing student performance.

This system will be a helpful tool for teachers in schools, making their job of managing student data much easier. Future improvements could include adding report generation and more advanced data analysis features.

**System Design**

1. **Architecture**

The architecture of the Student Management System is designed to be simple, modular, and scalable. It follows a client-server model where the client interface is developed using Tkinter, and the server side involves database operations handled by MySQL. The interaction between the client and the database ensures secure and efficient data management.

1. **Components**

Client Interface: Developed using Tkinter, providing a graphical interface for user interactions.

Database Server: MySQL database server for storing and managing student data.

Backend Logic: Python scripts handle the communication between the interface and the database.

1. **Data Flow**

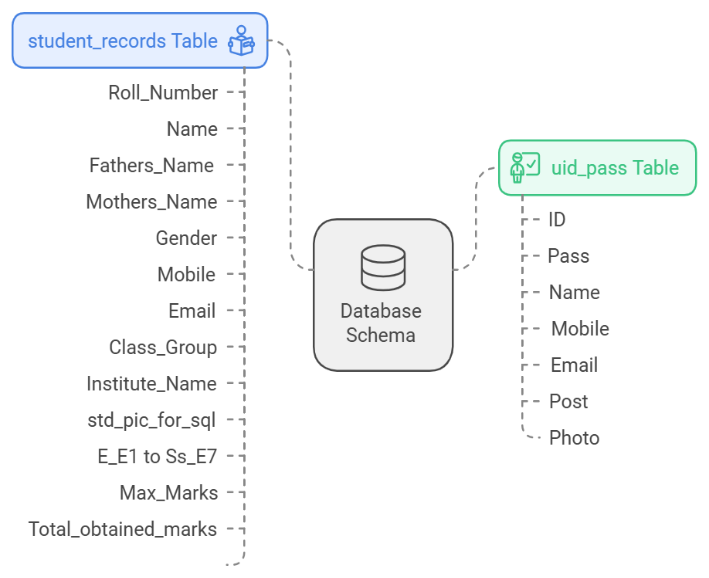
* User Input: Teachers interact with the GUI to manage student records.
* Processing: Python scripts process user inputs, perform CRUD operations, and handle business logic.
* Database Interaction: The processed data is sent to the MySQL database for storage or retrieval.
* User Feedback: The results of the operations are displayed back on the GUI for user confirmation.

1. **Database Design**

The database consists of two main tables: student\_records and teacher\_auth.

1. **student\_records Table**:

* Roll\_Number: A unique identifier for each student.
* Name, Fathers\_Name, Mothers\_Name: Personal information about the student.
* Gender, Mobile, Email: Contact details.
* Class\_Group: The class and group the student belongs to.
* Institute\_Name: The name of the school/institute.
* std\_pic\_for\_sql: A binary field for storing student photos.
* E\_E1 to Ss\_E7: Marks for various subjects and exams.
* Max\_Marks, Total\_obtained\_marks, Percentage: Academic performance metrics.

2. **uid\_pass Table**:

* ID: Unique identifier for each teacher.
* Pass: Password for teacher login.
* Name: Teacher’s full name.
* Mobile: Contact phone number.
* Email: Contact email address.
* Post: Position or role of the teacher.
* Photo: Binary data for storing teacher photos.

1. **Visual Representation of Database**

Student Records Table

| **Column Name** | **Description** |
| --- | --- |
| Roll\_Number | Unique identifier for each student. |
| Name | Full name of the student. |
| Fathers\_Name | Student's father's name. |
| Mothers\_Name | Student's mother's name. |
| Gender | Gender of the student. |
| Mobile | Mobile number of the student. |
| Email | Email address of the student. |
| Class\_Group | The class and group the student is in. |
| Institute\_Name | Name of the school/institute. |
| std\_pic\_for\_sql | Binary data for storing student photos. |
| E\_E1 to Ss\_E7 | Marks for various subjects/exams. |
| Max\_Marks | Maximum marks possible. |
| Total\_obtained\_marks | Total marks obtained by the student. |
| Percentage | Percentage of marks. |

Teacher Authentication Table

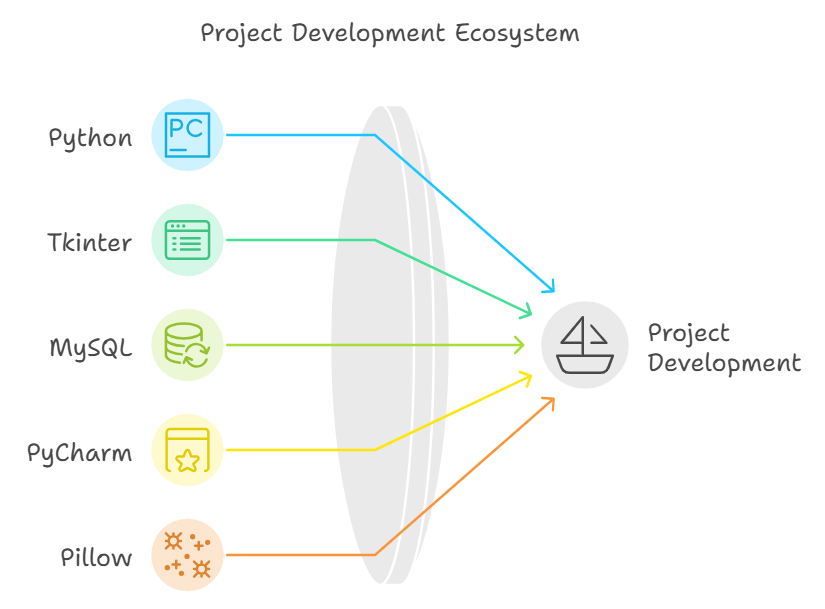
| **Column Name** | **Description** |
| --- | --- |
| ID | Unique identifier for each teacher. |
| Pass | Password for teacher login. |
| Name | Teacher’s full name. |
| Mobile | Contact phone number. |
| Email | Contact email address. |
| Post | Position or role of the teacher. |
| Photo | Binary data for storing teacher photos. |

**Implementation**

1. **Tools and Technologies**

This project leverages several tools and technologies to achieve its objectives:

* **Python**: The primary programming language used for building the application.
* **Tkinter**: A standard GUI library in Python used for creating the graphical user interface.
* **MySQL**: A relational database management system used for storing and managing student and teacher data.
* **PyCharm**: An integrated development environment (IDE) used for writing and managing the code.
* **Pillow**: A Python Imaging Library (PIL) fork used for image processing tasks.



1. **Modules and Functionality**

The application is divided into several modules, each responsible for specific functionality:

**1.**[**main.py**](https://main.py/?form=MG0AV3)

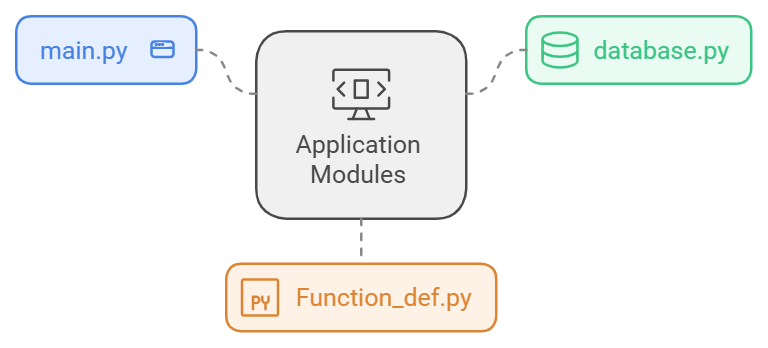
* **Description**: The entry point of the application. It initializes the main window, sets up the menu, and handles navigation between different frames.
* **Key Functions**:
  + \_\_init\_\_: Sets up the main window, menu, and initial frame.
  + login: Handles teacher login and redirects to the dashboard upon successful authentication.
  + dashboard: Displays options for managing student records and results.

**2.**[**database.py**](https://database.py/?form=MG0AV3)

* **Description**: Handles all database interactions, including connecting to MySQL and performing CRUD operations.
* **Key Functions**:
  + connect: Establishes a connection to the MySQL database.
  + fetch\_teach: Retrieves teacher data from the database based on login credentials.
  + fetch\_std: Retrieves student data based on search criteria.
  + insert\_std, update\_std, delete\_std: Manage student records in the database.

**3. Function\_def.py**

* **Description**: Contains auxiliary functions used throughout the application.
* **Key Functions**:
  + convert\_to\_binary: Converts images to binary format for storage in the database.
  + convert\_to\_og: Converts binary data back to image files for display.
  + clear\_entries: Clears input fields in the forms.



1. **User Interface**

The user interface is designed using Tkinter and is structured into multiple frames, each handling different functionalities:

**1. Login Frame**

* **Description**: Allows teachers to securely log in.
* **Key Elements**:
  + Username and password fields.
  + Login button that validates credentials and redirects to the dashboard.

**2. Dashboard Frame**

* **Description**: Serves as the main hub for navigation.
* **Key Elements**:
  + Buttons for navigating to student records, marks entry, and result visualization.

**3. Student Records Frame**

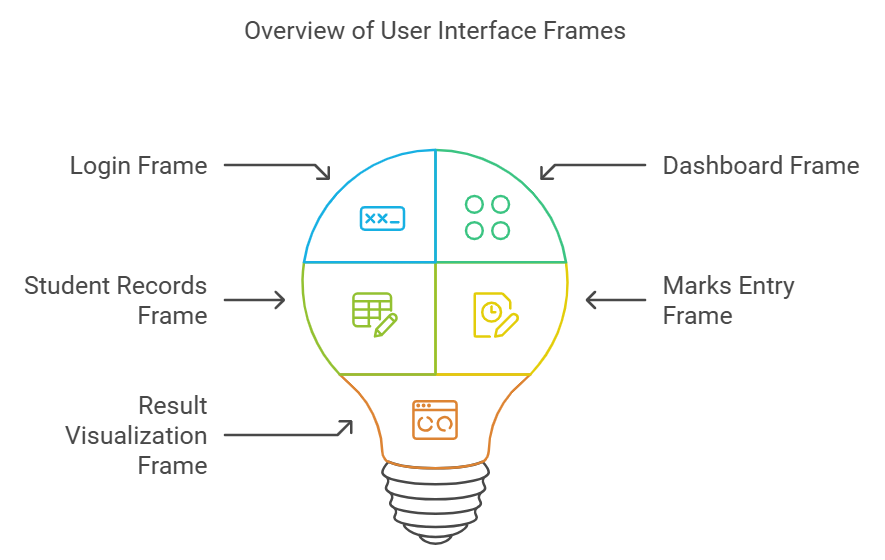
* **Description**: Enables CRUD operations on student records.
* **Key Elements**:
  + Fields for entering student details.
  + Buttons for adding, updating, viewing, and deleting records.

**4. Marks Entry Frame**

* **Description**: Facilitates entering and updating student marks.
* **Key Elements**:
  + Fields for entering marks for different subjects.
  + Buttons for saving and updating marks.

**5. Result Visualization Frame**

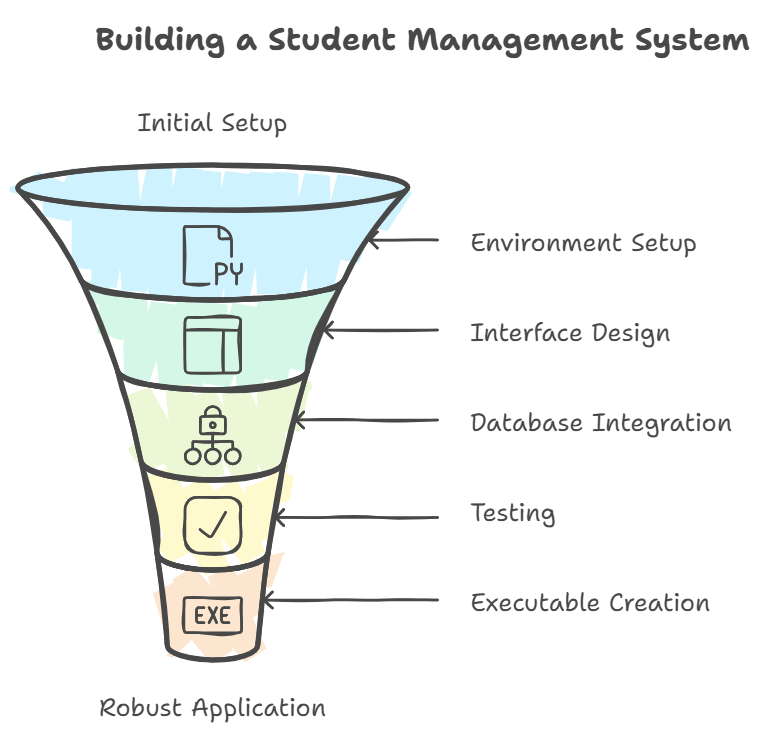
* **Description**: Displays and visualizes student performance.
* **Key Elements**:
  + Graphical representation of marks.
  + Options to filter and view specific results.



1. **Implementation Steps**
2. **Setting Up the Environment**:
   * Install Python and required libraries.
   * Set up MySQL database and create necessary tables.
   * Configure PyCharm for project development.
3. **Building the Interface**:
   * Design the layout using Tkinter.
   * Implement navigation between different frames.
   * Add input validation and error handling.
4. **Integrating the Database**:
   * Establish a connection to the MySQL database.
   * Implement functions for CRUD operations.
   * Securely store and retrieve data.
5. **Testing**:
   * Test individual modules for functionality and performance.
   * Conduct integration testing to ensure smooth operation of the complete system.
   * Fix any bugs and optimize performance.
6. **Creating the Executable**:
   * Use PyInstaller to package the application into an executable file.
   * Ensure all dependencies and resources are included.

This detailed implementation plan outlines how the Student Management System was built,

ensuring a robust and user-friendly application for managing student data.



**Testing**

1. **Test Cases**

Testing is a critical phase in the software development process. It ensures that the application functions correctly and meets the intended requirements. Here are the key test cases conducted for the Student Management System:

1. Login Functionality

* Test Case: Verify teacher login with correct credentials.
  + Input: Correct username and password.
  + Expected Outcome: User should be redirected to the dashboard.
  + Result: Pass
* Test Case: Verify login with incorrect credentials.
  + Input: Incorrect username or password.
  + Expected Outcome: Error message should be displayed.
  + Result: Pass

2. Add Student Record

* Test Case: Verify adding a new student record.
  + Input: Enter valid student details and submit.
  + Expected Outcome: Student record should be saved in the database.
  + Result: Pass

3. Update Student Record

* Test Case: Verify updating an existing student record.
  + Input: Modify student details and save.
  + Expected Outcome: Student record should be updated in the database.
  + Result: Pass

4. Delete Student Record

* Test Case: Verify deleting a student record.
  + Input: Select a student record and delete.
  + Expected Outcome: Student record should be removed from the database.
  + Result: Pass

5. View Student Record

* Test Case: Verify viewing student records.
  + Input: Search for a student by Roll Number or Name.
  + Expected Outcome: Student details should be displayed.
  + Result: Pass

6. Marks Entry

* Test Case: Verify entering marks for a student.
  + Input: Enter marks for different subjects and submit.
  + Expected Outcome: Marks should be saved and associated with the student.
  + Result: Pass

7. Result Visualization

* Test Case: Verify visualization of student results.
  + Input: Select a student to view marks.
  + Expected Outcome: Marks should be displayed in graphical format.
  + Result: Pass

1. **Results and Analysis**

The testing phase demonstrated that the Student Management System meets the intended functionality. All critical test cases passed successfully. The system was tested for various scenarios, ensuring robust performance and reliability.

**Key Findings:**

* **Login Module**: Handled both successful and unsuccessful login attempts efficiently, providing appropriate feedback.
* **CRUD Operations**: Add, update, delete, and view functionalities for student records worked seamlessly, ensuring data integrity.
* **Marks Management**: Accurate entry and display of student marks, with no discrepancies.
* **Result Visualization**: Displayed student marks effectively, providing clear and intuitive visual feedback.

**Bug Fixes and Optimizations:** During the testing process, a few minor bugs were identified and promptly fixed. These included:

* **Input Validation**: Ensured all required fields were filled out before submission.
* **Error Handling**: Improved error messages for invalid operations.
* **Performance Optimization**: Enhanced database query performance for faster data retrieval.

The successful testing phase confirms the reliability and efficiency of the Student Management System, making it a valuable tool for teachers in managing student data.

**Conclusion**

The development of the Student Management System for teachers has proven to be a valuable and practical solution for managing student data efficiently. This project integrates modern technologies such as Python's Tkinter for the graphical user interface and MySQL for robust data storage, ensuring a seamless and user-friendly experience for teachers.

**Summary**

The primary objectives of the project were successfully achieved. An intuitive interface was developed, allowing teachers to manage student records with ease. The implementation of a secure login system ensures that only authorized personnel can access and manage the data. The functionalities for adding, updating, viewing, and deleting student records were implemented effectively, enhancing the overall efficiency of data management in educational institutions.

**References**

1. Books and Articles:
   * Smith, John. *Database Management Systems*. 3rd ed., McGraw-Hill, 2008.
2. Websites and Online Resources:
   * Python Software Foundation. "Tkinter - Python Interface to Tcl/Tk." [*Python.org*](https://python.org/?form=MG0AV3), Python Software Foundation, [https://docs.python.org/3/library/tkinter.html.](https://docs.python.org/3/library/tkinter.html?form=MG0AV3)Accessed 10 Oct. 2024.
   * Oracle Corporation. "MySQL 8.0 Reference Manual." *MySQL Documentation*, Oracle Corporation, [https://dev.mysql.com/doc/refman/8.0/en/.](https://dev.mysql.com/doc/refman/8.0/en/?form=MG0AV3)Accessed 15 Oct. 2024.
   * PyCharm Documentation. "Getting Started with PyCharm." *JetBrains*, [https://www.jetbrains.com/pycharm/documentation/.](https://www.jetbrains.com/pycharm/documentation/?form=MG0AV3)Accessed 5 Oct. 2024.
3. Software and Tools:
   * PyInstaller. "PyInstaller Documentation." *PyInstaller Manual*, [https://pyinstaller.readthedocs.io/en/stable/.](https://pyinstaller.readthedocs.io/en/stable/?form=MG0AV3)Accessed 12 Oct. 2024.
   * Pillow Documentation. "Python Imaging Library (PIL) Fork." *Read the Docs*, [https://pillow.readthedocs.io/en/stable/.](https://pillow.readthedocs.io/en/stable/?form=MG0AV3)Accessed 7 Oct. 2024.
   * PyAutoGUI. "PyAutoGUI Documentation." PyAutoGUI Library, <https://pyautogui.readthedocs.io/en/latest/> . Accessed 11 Oct. 2024.
4. Python Libraries and Modules:
   * *MySQL Connector/Python Documentation*. [https://dev.mysql.com/doc/connector-python/en/.](https://dev.mysql.com/doc/connector-python/en/?form=MG0AV3)Accessed 8 Oct. 2024.
   * Cannon, Brett. *Pillow (PIL Fork) Documentation*. [https://pillow.readthedocs.io/en/stable/.](https://pillow.readthedocs.io/en/stable/?form=MG0AV3)Accessed 7 Oct. 2024.

**Appendices**

**Appendix A: Source Code**

* **main.py**

<https://docs.google.com/document/d/1iOckxkS7m0g7L9W6MXhdoaBoCjvqdq3LQhrOqLs2F2g/edit?usp=sharing>

* **function\_def.py**

<https://docs.google.com/document/d/1Be8FLtr_2ahuW6tQsgKH7s-b37Qnvp7v3Et-gCBMsm4/edit?usp=sharing>

* **database.py**

<https://docs.google.com/document/d/1bj_MSQdLiyiH1ZMUtKDw2AMvuk58F_T_5CP2TqeliA0/edit?usp=sharing>

**Demo**: <https://docs.google.com/document/d/1_OumVMFp_686vgSlvL9DFZ-U2wTgRpRZWMufa9Phkeo/edit?usp=sharing>