**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction to DBMS**

Database is a collection of related data. DBMS came into existence in 1960 by Charles. Again in 1960 IBM brought IMS-Information management system. In 1970 EdgorCodd at IBM came with new database called RDBMS. In 1980 then came SQL Architecture- Structure Query Language. In 1980 to 1990 there were advances in DBMS e.g. DB2, ORACLE. A database has the following implicit properties:

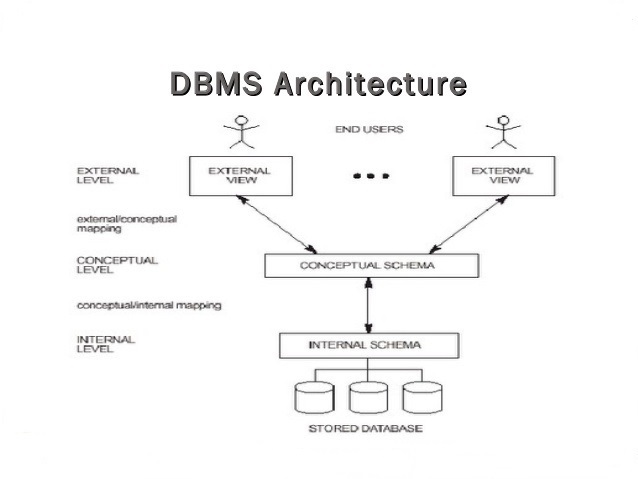
* A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
* A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
* A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents.

Metadata (meta data, or sometimes meta information) is "data about data", of any sort in any media. An item of metadata may describe a collection of data including multiple content items and hierarchical levels, for example a database schema. In data processing, metadata is definitional data that provides information about or documentation of other data managed within an application or environment. The term should be used with caution as all data is about something, and is therefore metadata.

A database management system (DBMS) is a collection of programs that enables users to create and maintain database. The DBMS is a general purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.

Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database. The descriptive information is also stored in the database in the form database catalogue or dictionary; it is called meta-data. Manipulating the data includes the querying the database to retrieve the specific data. An application program accesses the database by sending the queries or requests for data to DBMS. The important function provided by the DBMS includes protecting the database and maintain the database.



**Figure 1.1: Three schema architecture**

* 1. **Overview of the project**

The “Online voting System” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by the existing system. Moreover, this system is designed for a particular need of the institution to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error messages while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it provides it is user-friendly. This approach is usually adopted with a view to decrease the percentage of the voting problems. Thus, it will help Institution in better utilization of resources.

* + 1. **Problem statement:**

To allocate the slot to students and maintain an updated information about voting results.

* + 1. **Objectives of the project:**
* The purpose of this is to build an application to reduce the manual work for managing elections in colleges.
* It tracks all the details about the results of voting system.
* It maintains the count of votes casted by voters for each party
* The main objective behind this project is to provide voting facilities for students of the Institution
* By using this system, the elections can be held in a systematic manner without any electoral fraud.

**CHAPTER 2**

**SYSTEM DESIGN AND METHODOLOGY**

**2.1. System Architecture**

**USER**

**Login**

**Vote**

**Result**

**Database**

**Figure 2.1: System Architecture of online voting System**

The Figure 2.1 describes the System Architecture of a online voting System. The architecture consists of a centralized database, which will be accessed by only one type of users namely: user. user access is required for the security, which is implemented through login module with which the security can login with their registered userid.

Once login is successful, they can give their respective vote to a candidate of any party of their choice, they can also see the overall percentage of votes received by each candidate.

**2.2. ER Diagrams**

Student register

Vote result

Votes given

**Figure 2.2: ER Diagram of Online Voting System**

An Entity-Relationship Diagram (ERD) is a data modelling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The above ER Diagram consists of 2 entities namely

1. Student register
2. Vote result

The student register consists of the following attributes: usn, email, dept, voting, student name, voterid. Here the usn is the primary key.

The vote result consists of the following attributes: Si\_no, party name, colour, total\_vote\_given. Here party name is the primary key.

**2.3 Schema Diagram**

Usn email year\_of\_paassing dept voting stu\_name voter\_id

Si\_no party\_name colour total\_vote\_given

**Figure 2.3: Schema Diagram of Online voting System**

**CHAPTER 3**

**MODULE IMPLEMENTATION**

**3.1. Platform Description**

**SQL Lite as a Back End**

SQLite is a C-language library that implements a [small](https://www.sqlite.org/footprint.html), [fast](https://www.sqlite.org/fasterthanfs.html), [self-contained](https://www.sqlite.org/selfcontained.html), [high-reliability](https://www.sqlite.org/hirely.html), [full-featured](https://www.sqlite.org/fullsql.html), SQL database engine. SQLite is the [most used](https://www.sqlite.org/mostdeployed.html) database engine in the world. Full featured SQL implementation with advanced capabilities like [partial indexes](https://www.sqlite.org/partialindex.html), [indexes on expressions](https://www.sqlite.org/expridx.html), [JSON](https://www.sqlite.org/json1.html), [common table expressions](https://www.sqlite.org/lang_with.html), and [window functions](https://www.sqlite.org/windowfunctions.html). ([Omitted features](https://www.sqlite.org/omitted.html)). SQL Lite is a Cross-platform(X): Android, \*BSD, iOS, Linux, Mac, Solaris, VxWorks, and Windows (Win32, WinCE, WinRT) are supported out of the box. Easy to port to other systems. It is written in ANSI-C TCL binding included. Bindings for dozens of other languages available separately. Since most actual web server deployments use the same components as SQL Lite, it makes transitioning from a local test server to a live server extremely easy as well.

**Python Tkinter as a Front-End**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the *Tkinter* module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

**3.2. Module Description**

**1. REGISTERATION   
INPUT**

* User-Name, Usn, Email, Year of passing, department

**OUTPUT:**

A successful register, shall take the user to the respective login page, else if the register attempt fails, an error message will be displayed.

**DESCRIPTION:**

Register page provides textboxes to enter name, usn, Email, year of passing, department. Enter the credentials and click on submit button. If the user had entered all the details correctly without leaving any empty textboxes then an voterid is sent to his email id. if any textbox is left empty then an error message is displayed.

On successful registration, the user will be allocation with a particular Account is created.

**LOGIN**

**INPUT**

For logging in the user has to enter the voterId which had been received by Email.

**OUTPUT:**

User will be logged in successfully.

**DESCRIPTION:**

When the user enters the correct password then he will get access to vote for any party of his interest. He will be directed to the voting page. If any user tries to login twice then an error message is displayed.

1. **VOTING PAGE**

**INPUT**

Give vote to a party

**OUTPUT:**

Vote successful page will be displayed

**DESCRIPTION:**

After the user enrols his vote, the user will be directed to vote successful page. when he enters ok then again login page will be displayed.

1. **RESULT PAGE**

**INPUT**

Enter Result textbox in login page

**OUTPUT**

Percentage of votes received by each party will be displayed.

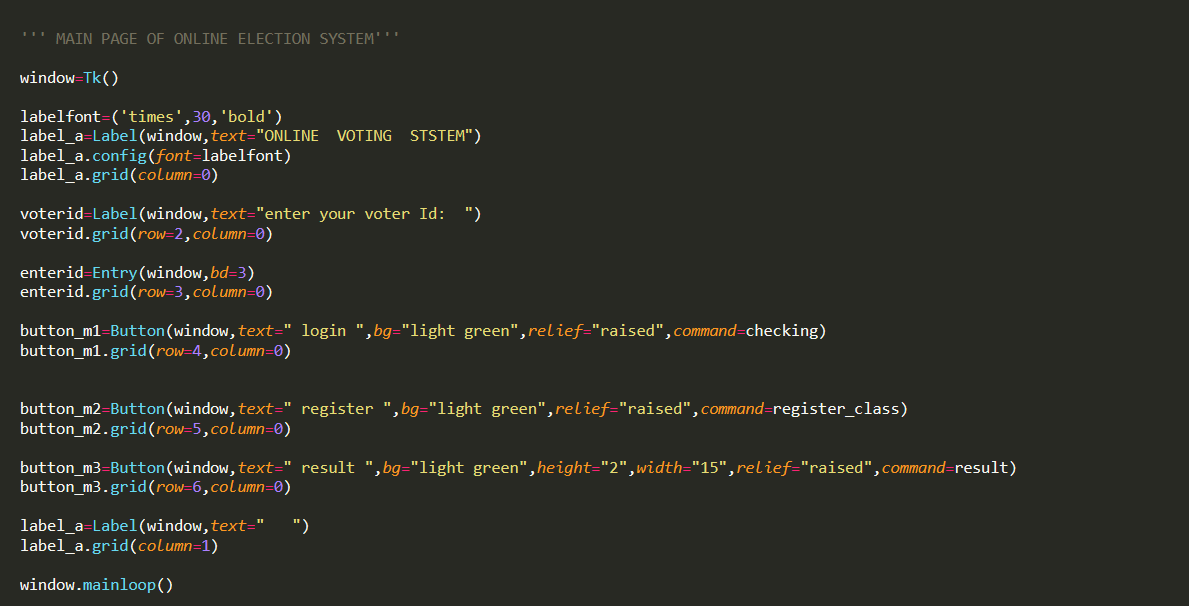
**DESCRIPTION:**

When user enter the result textbox then he will be directed to the result page where the percentage of votes received by each party will be displayed.

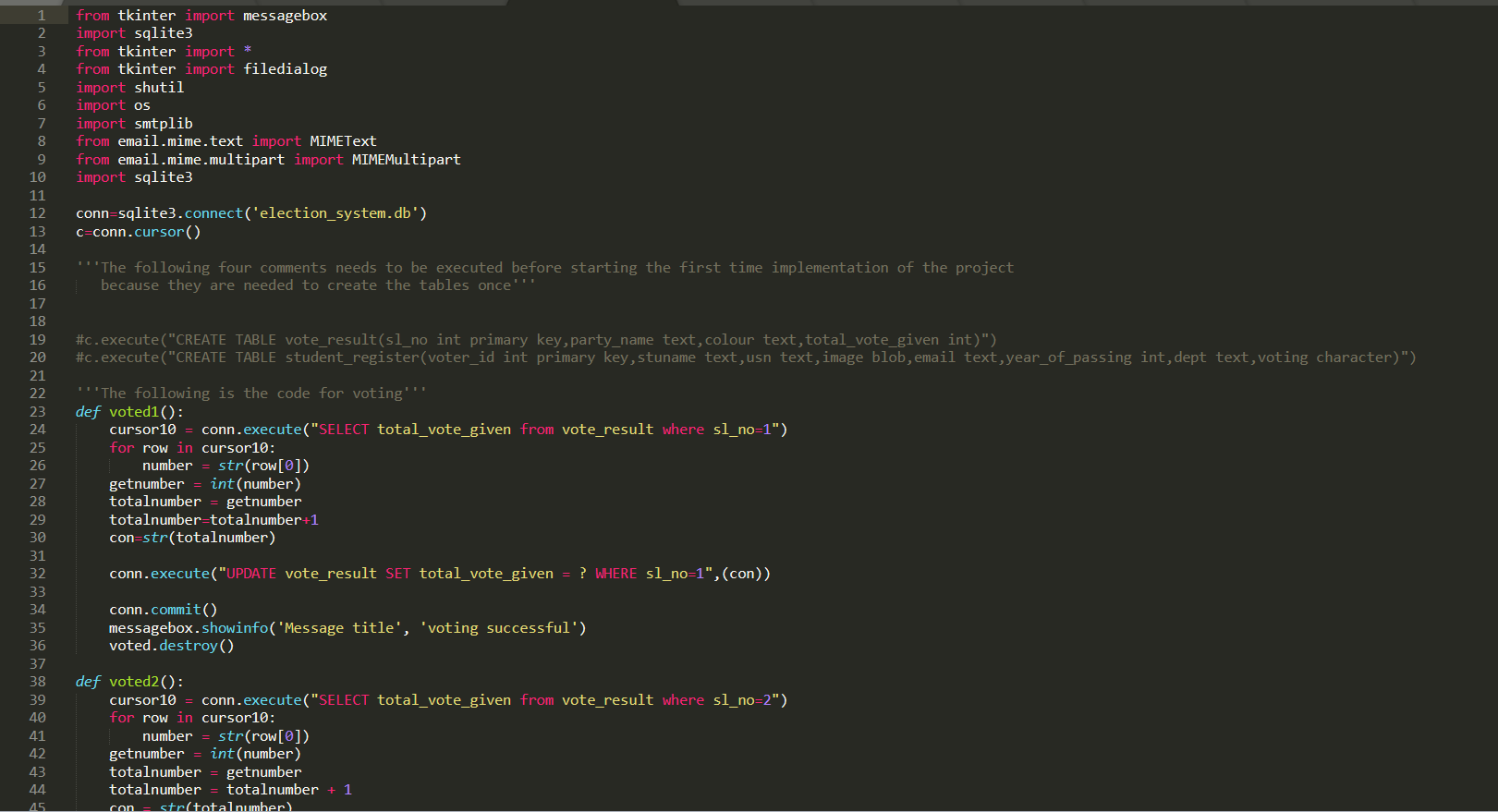
**CHAPTER 4**

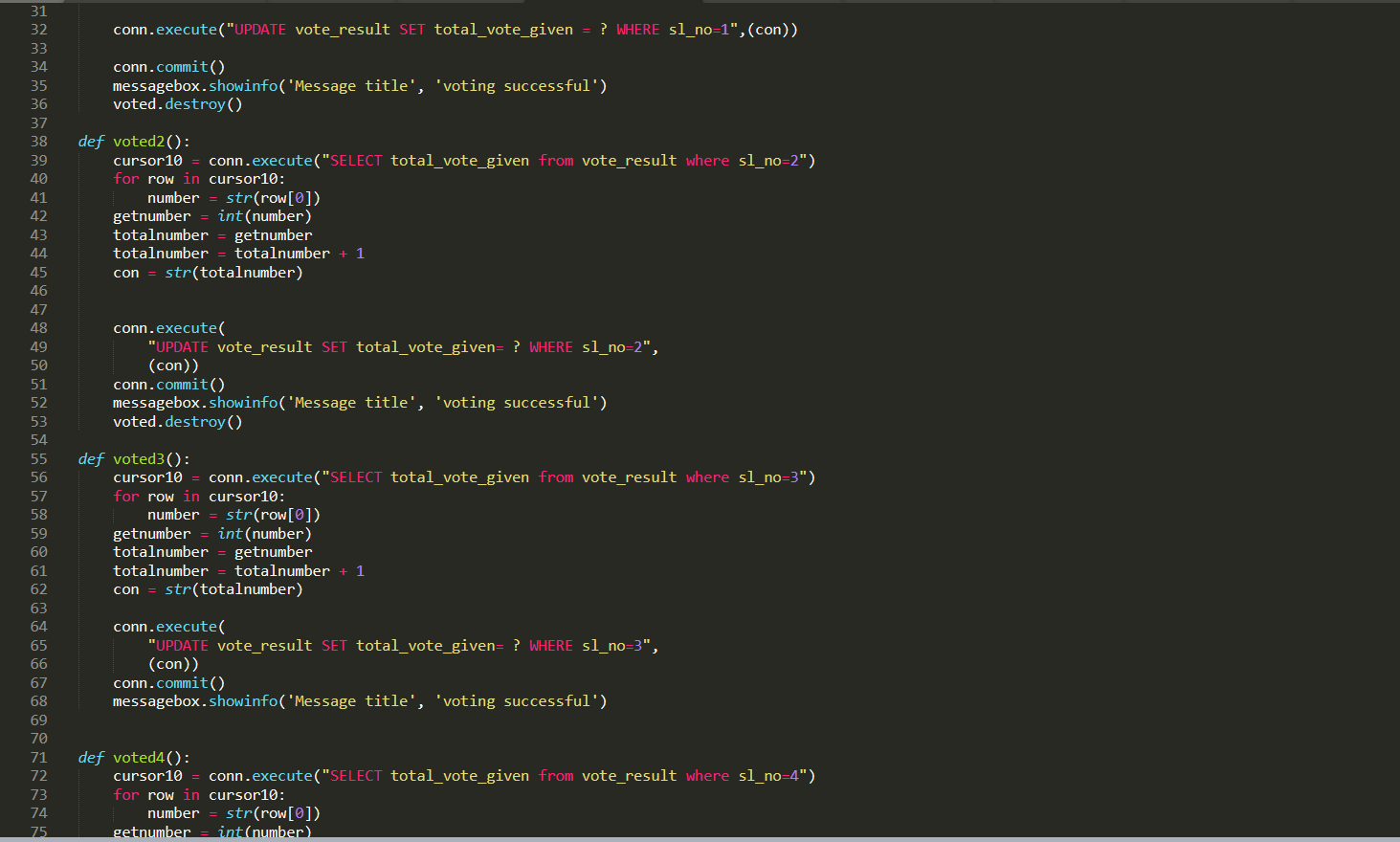
**CODING**

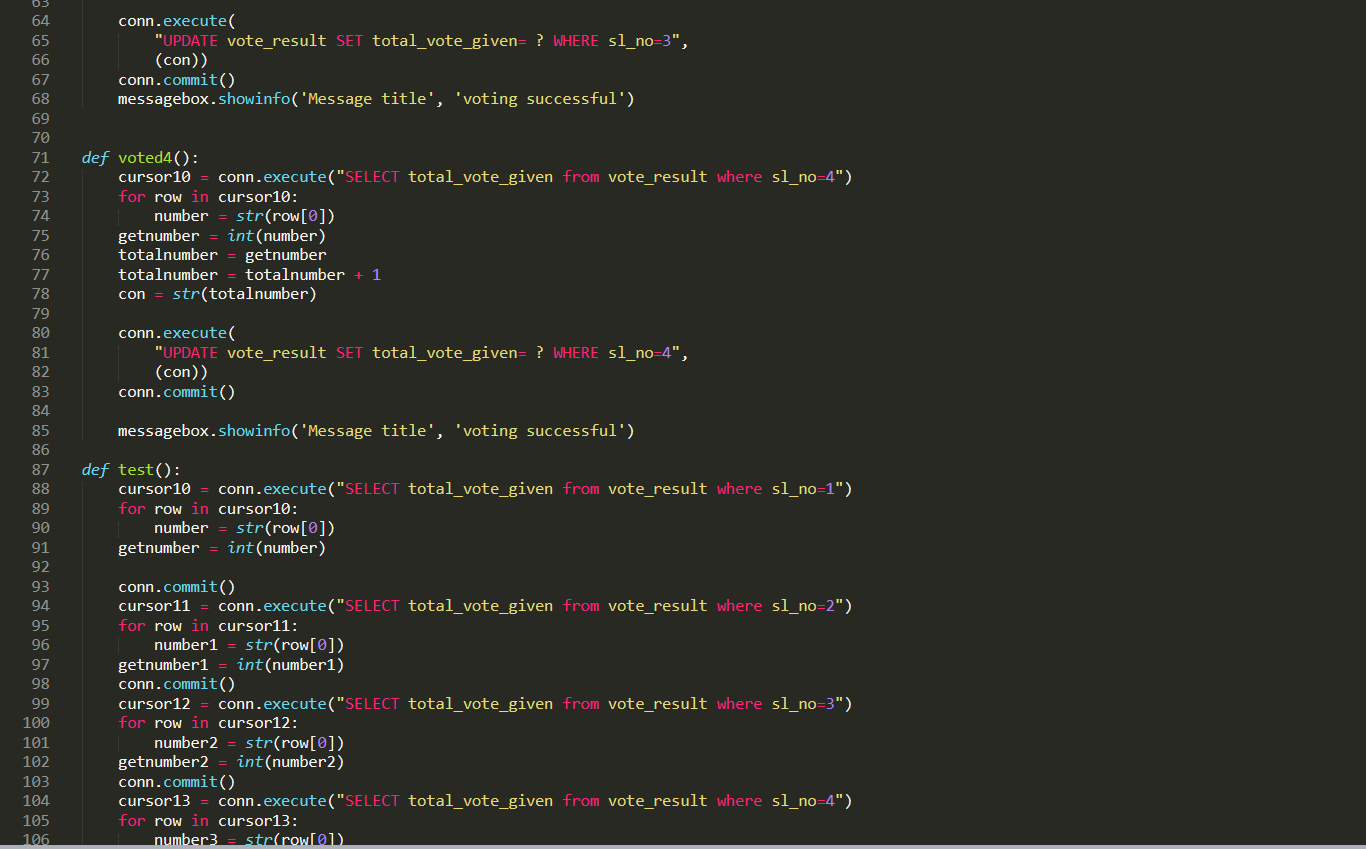
**Main page**

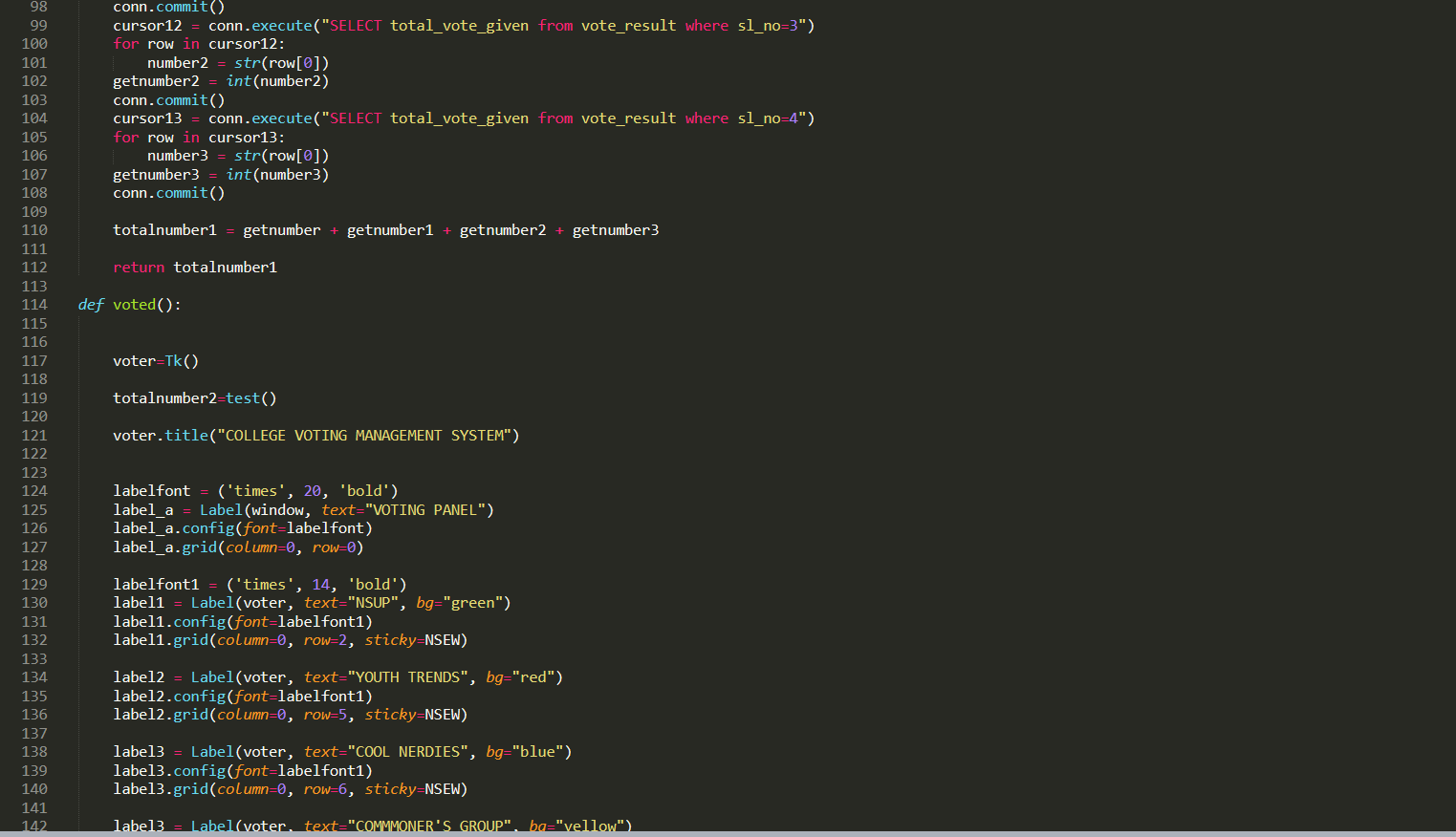


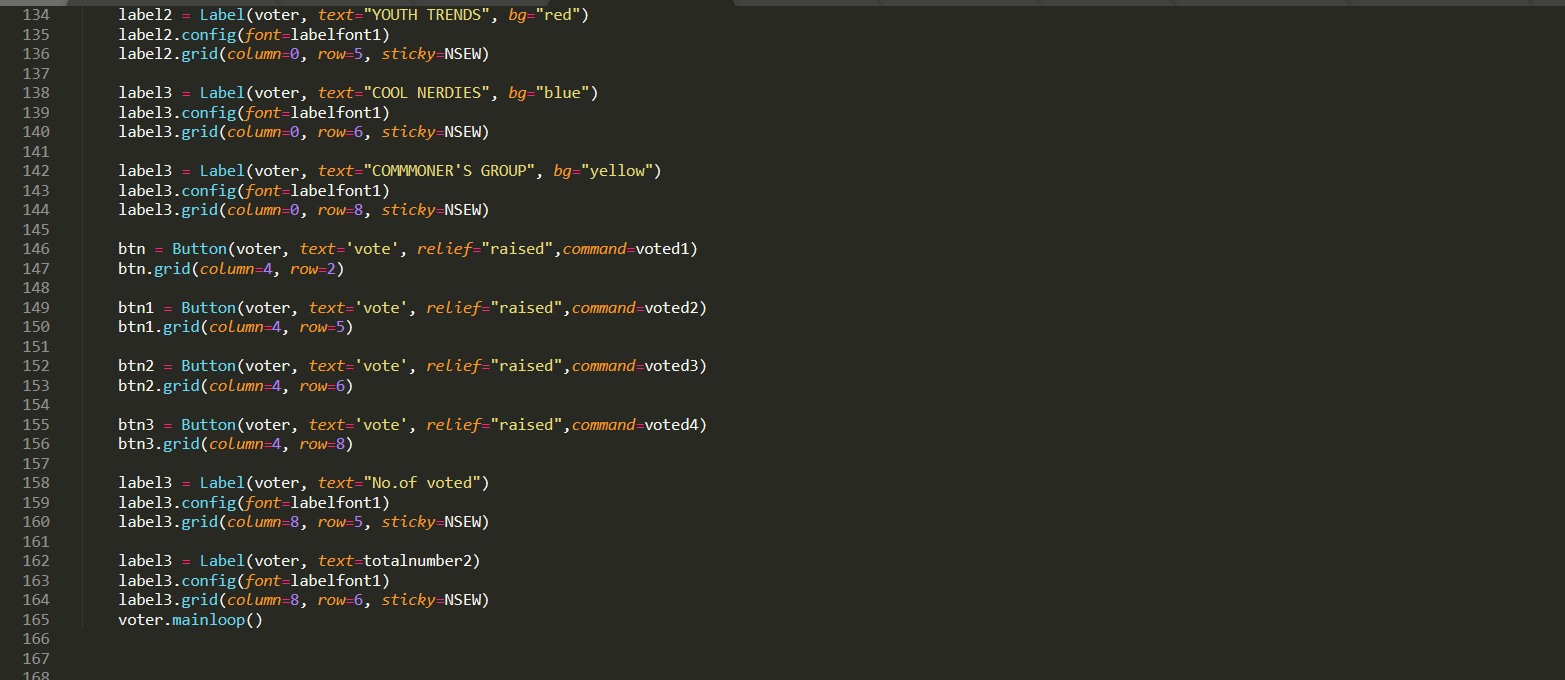
**Voting page**

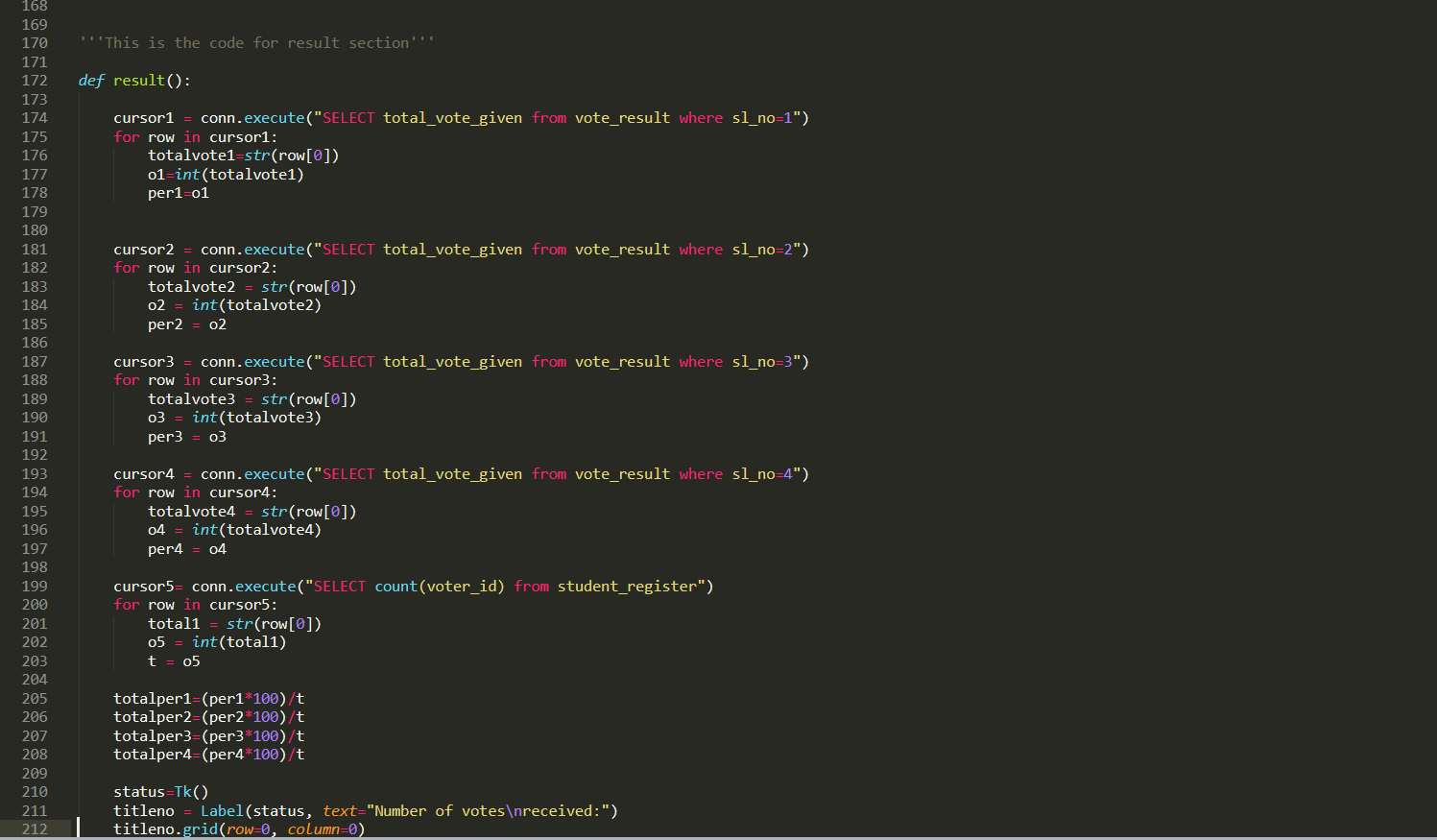




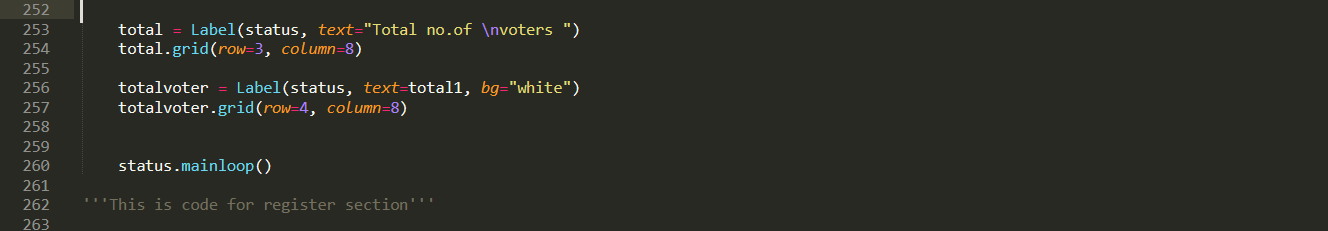




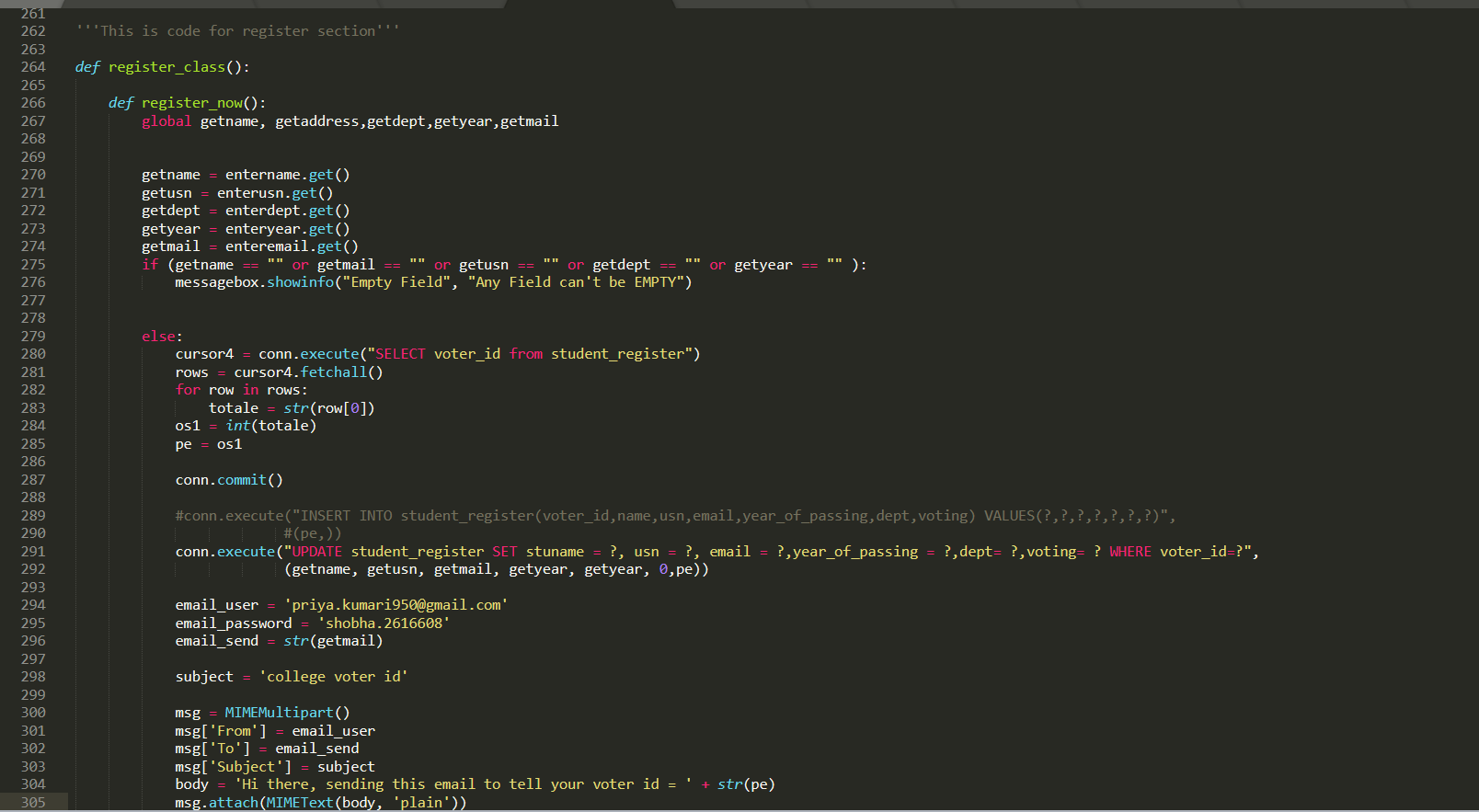


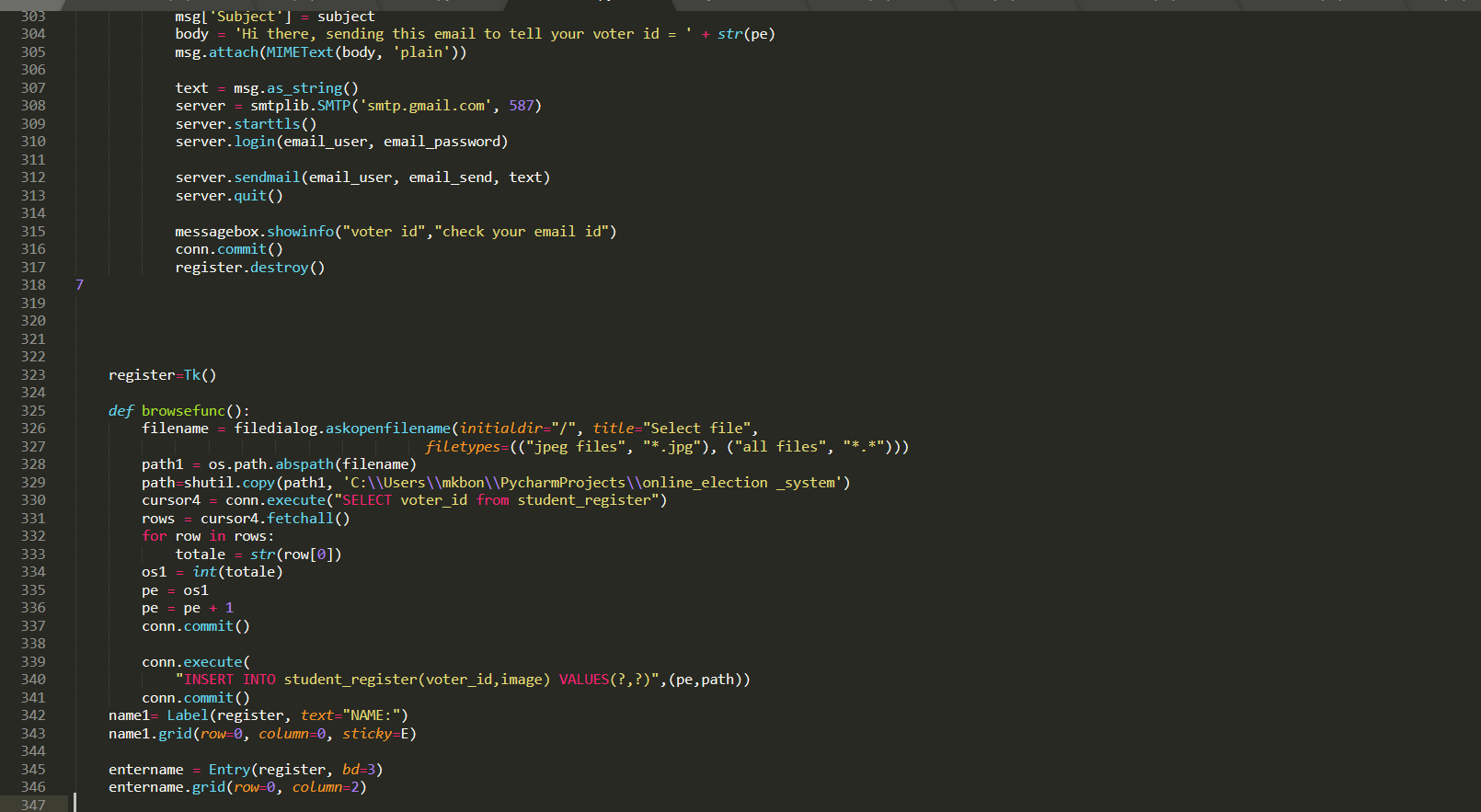
**Result page**

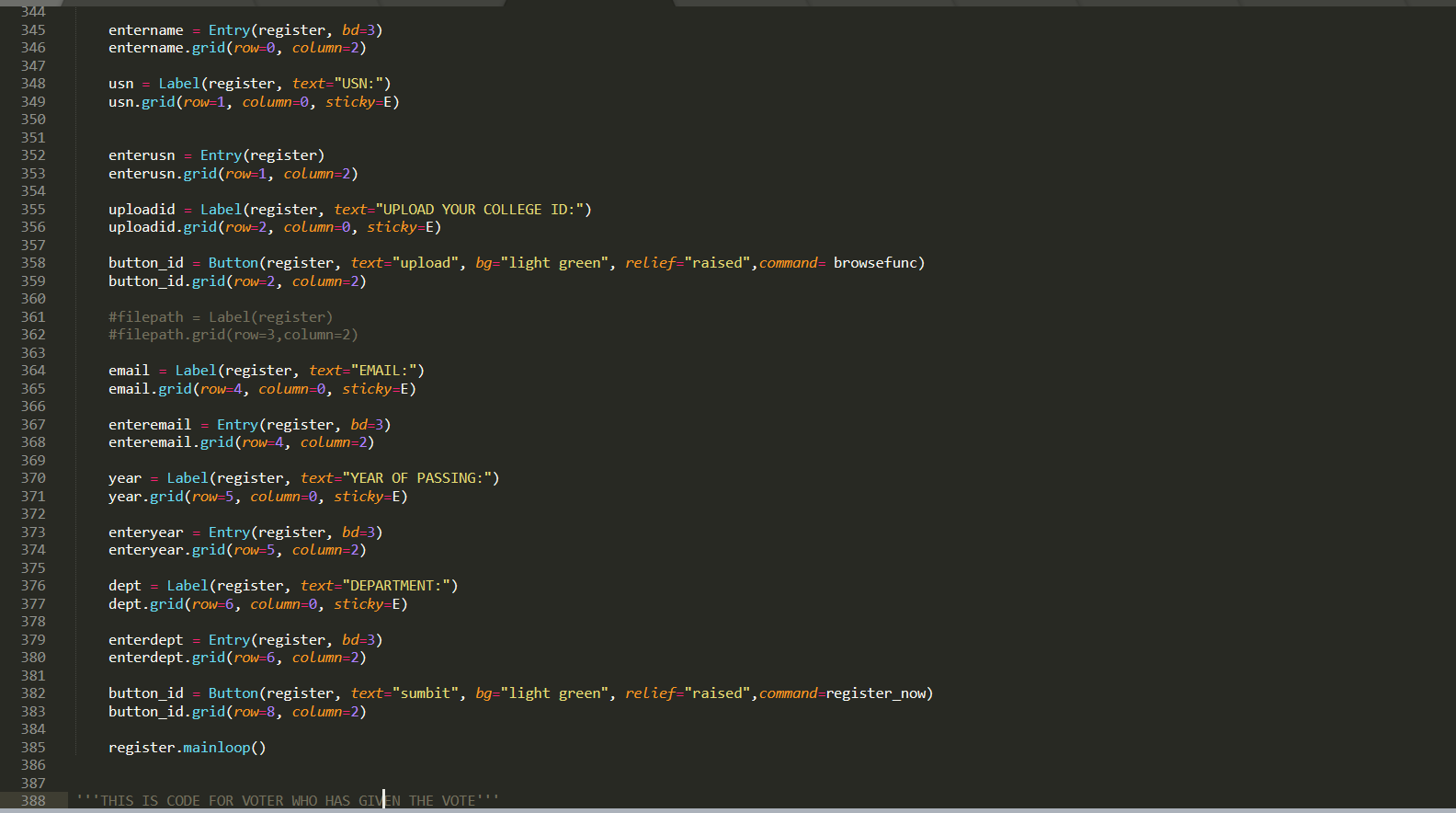




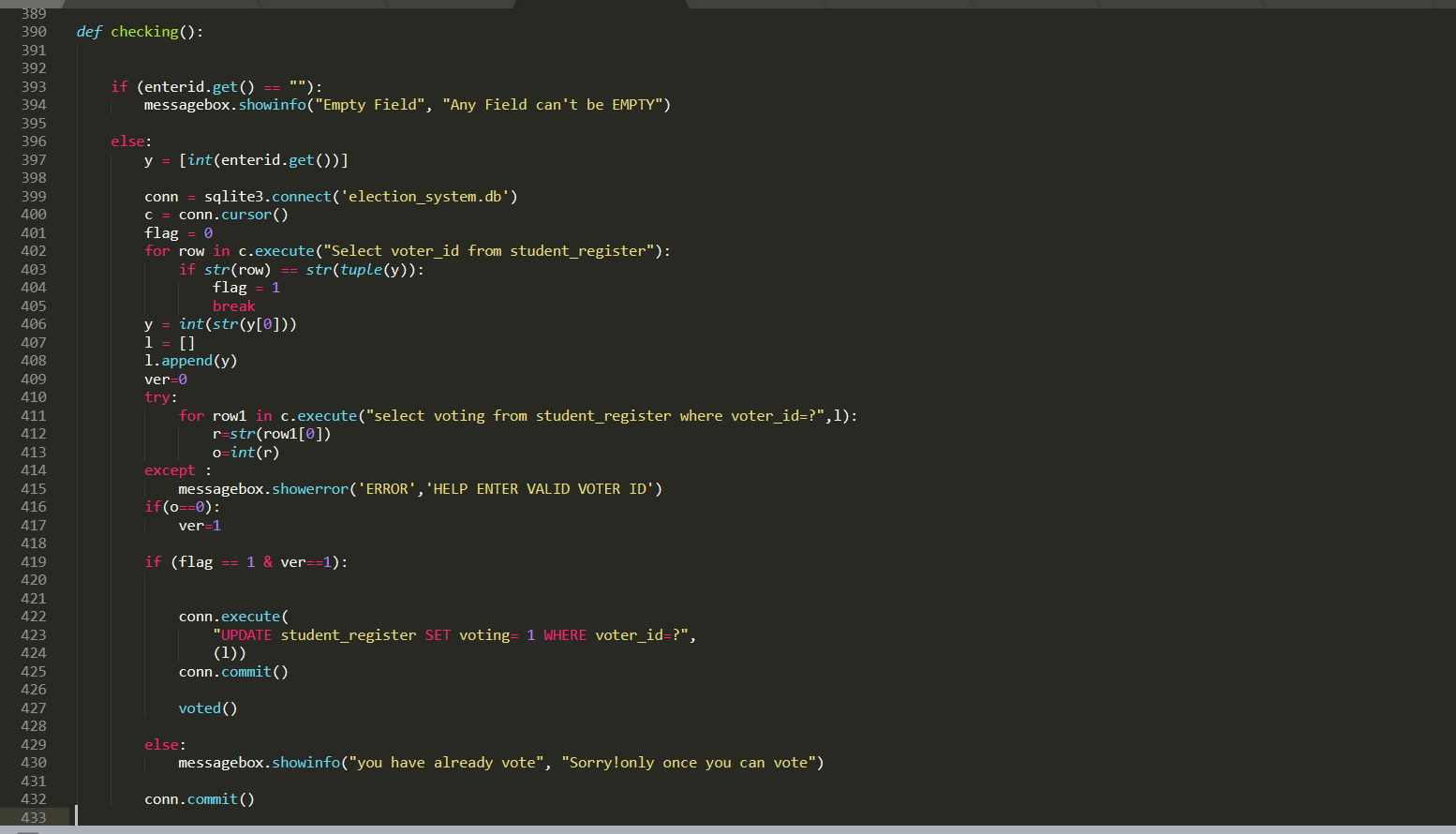
**Register page**







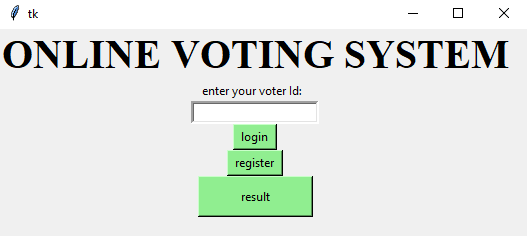
**Vote Successful page**



**CHAPTER 5**

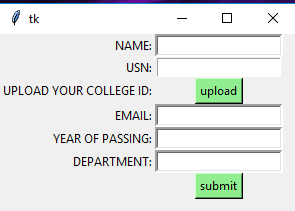
**SCREENSHOTS**

**5.1. MAIN PAGE**

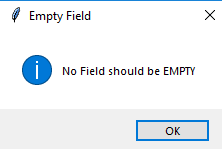


**Figure: 5.1: Online Voting System Main Page**

**5.2 Registration Page**



**Figure: 5.2: Registration Page**

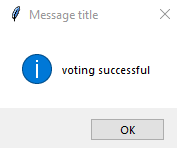


**Figure: 5.2.1: Error Page**

**5.3 Voting Page**

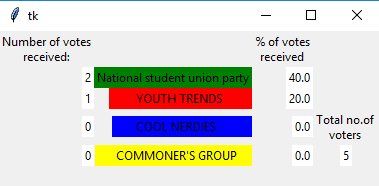


**Figure: 5.4: Login Page**



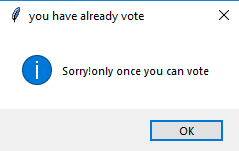
**Figure: 5.3.1: Voting successful Page**

**5.4 Result Page**



**Figure: 5.4: Result Page**

**5.5 Login Error Page**



**Figure: 5.5: Login Error Page**

**CHAPTER 6**

**CONCLUSION AND FUTURE WORKS**

**Conclusion:**

Elections play one of the major role in our day to day lives. By using online election system voters would have to be able to ensure that their vote has been counted as cast, but the system should still not produce a receipt that could be used to pressure voters or sell votes.

A separate recount of votes to detect and correct mistakes that have occurred in the vote count is not possible in online voting. In an online voting system, the information produced by the system on which votes have been counted needs to be trusted. If a recount of votes was carried out in an online voting system, it would be based on the information produced by the system itself and would give the same result as the actual count. The Security Strategy for Society1 sets the following goal: "The operating conditions required for holding History has proven banks to be vulnerable to many risks, however, including credit, liquidity, market, operating, interesting rate and legal risks. Many global crises have been the result of such vulnerabilities and this has led to the strict regulation of state and national banks.

Hence the online voting system is helpful in a safe election without any attempt of electoral fraud. The user can access the Login portal from any of his systems and will have secured voterId which reduces the risk of duplicate voting.

**Future works:**

1. Providing sensor-based access to vote
2. We can also add face recognition system to confirm the voter

**BIBLIOGRAPHY**

[1] Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson

[2] [www.stackoverflow.com](http://www.stackoverflow.com)

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.scribd.com](http://www.scribd.com)

[5] [www.tutorialpoint.com](http://www.tutorialpoint.com)

[6] [www.roseindia.net](http://www.roseindia.net)

[7] [www.slideshare.net](http://www.slideshare.net)

[8] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition 2014, McGraw Hill

[9] Charles R. Severance, “Python for Everybody: Exploring Data Using Python 3”, 1st Edition, CreateSpace Independent Publishing Platform, 2016.

[10] Allen B. Downey, "Think Python: How to Think Like a Computer Scientist”, 2 ndEdition, Green Tea Press, 2015.