PROJECT

SYNOPSIS

SYNOPSIS

1. **Project Title:**

College Voting System

1. **Introduction and objectives of the project:**

This project is aimed at developing a College Voting System for Colleges. The main objective of developing this system is for voting purpose which saves lot of time in counting process. It will make voting process fully computerized which is very fast and efficient. Also this web application maintains college student’s record, candidate record and voting record. This software can be used by any colleges to make election computerized.

College Voting System offers many advantages over the conventional paper-based electoral systems, including mobility and convenience for voters, greater speed and accuracy in the counting process, prevention of involuntary voting errors, better accessibility, lower costs, support of multiple languages, greater flexibility, etc. It has several core functionalities: security, ease-of-use, accessibility, and scalability/flexibility.

1. **Project category:**

Web based Relational Database Management System (Web RDBMS)

1. **Tools and platform:**

**Hardware requirements**

● Processor: Intel dual core

● Operating system : Windows XP

● RAM : 512 MB

● Keyboard, Monitor, etc

**Software requirements**

● Software: XAMPP

● Server : Apache server

● Database server: MySQL server

● IDE: Adobe Dreamweaver or Notepad++

1. **Project modules:**

* **Student Registration module:** This module allows student to register by entering their roll number, name, contact number, address, etc. After the registration staff can verify registration details. After successful verification the students can login to system by entering login credentials.
* **Candidate module:** The admin can add candidate details for the voting. The candidate list displays during voting.
* **Voting module:** In the voting module students can vote for candidates. Students can vote during voting schedule by selecting candidate. Student’s needs to login to the system for voting.
* **Voting result module:** This page is the public page where everyone can view voting result, winners list, total votes received, etc. The system automatically calculates voting records.
* **Dashboard module:** The dashboard module is for administrator and college staff. Administrator has full authority of the website. Staff can view reports and they can access some features in the dashboard module.
* **Voting types module:** This module is available for administrator where admin can add different types of voting records.

1. **Limitations:**

* Weak cyber-security infrastructure thus whole voting process could be hacked.
* Data collection-collecting data from the stakeholders of the system and involved personnel can be a challenge if required authorization is not provided.

1. **Innovativeness and usefulness:**

* It increases the counting speed.
* It reduces the human effort.
* There is no chance of wastage of votes.
* It saves lot of time.
* Results will be announced within a short period of time.

1. **Future scope of project:**

* We can implement RFID based voting system / EVM system in future.
* SMS based voting result notification can be sent to the students.

PREAMBLE

Preamble

The main purpose of developing College Voting System for multiple Colleges. The main objective of developing this system is for voting purpose which saves lot of time in counting process. It will make voting process fully computerized which is very fast and efficient. Also this web application maintains college student’s record, candidates record and voting record. This software can be used by any colleges to make election computerized.

College Voting System offers many advantages over the conventional paper-based electoral systems, including mobility and convenience for voters, greater speed and accuracy in the counting process, prevention of involuntary voting errors, better accessibility, lower costs, support of multiple languages, greater flexibility, etc. It has several core functionalities: security, ease-of-use, accessibility, and scalability/flexibility.

SOFTWARE REQUIREMENT SPECIFICATION

SOFTWARE REQUIREMENT SPECIFICATION

1. **Introduction:**

A **software requirements specification** (SRS) is a description of a [software system](https://en.wikipedia.org/wiki/Software_system) to be developed, laying out [functional](https://en.wikipedia.org/wiki/Functional_requirement) and [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirements), and may include a set of [use cases](https://en.wikipedia.org/wiki/Use_case) that describe interactions the users will have with the software.

A basic purpose of the SRS is to bridge this communication gap between client and the developer so they have a shared vision of the software being built. An SRS establishes the basis for agreement between the client and the supplier on what the software product will do. SRS provides a reference for validation of the final product. A high-quality SRS is a pre-requisite to high-quality software and also reduces the development cost.

The introduction of the Software Requirement Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete “**College Voting System**”by defining the problem statement in detail. The detailed requirements of “**College Voting System**” are provided in this document.

* 1. **Purpose:**

This Document includes software requirements for the “**College Voting System**” Project. The purpose of this document is to detail the user requirements for all the functionality in “**College Voting System**”. This document is meant to serve as a guide to the developers and users. The purpose of the requirement document is to specify and provide all the information required to design, develop and test the system. This document ensures that the person reading the document understands what she/he is looking for.

The main purpose is developing a College Voting System for multiple Colleges. The main objective of developing this system is for voting purpose which saves lot of time in counting process. It will make voting process fully computerized which is very fast and efficient. Also this web application maintains college student’s record, candidates record and voting record. This software can be used by any colleges to make election computerized.

College Voting System offers many advantages over the conventional paper-based electoral systems, including mobility and convenience for voters, greater speed and accuracy in the counting process, prevention of involuntary voting errors, better accessibility, lower costs, support of multiple languages, greater flexibility, etc. It has several core functionalities: security, ease-of-use, accessibility, and scalability/flexibility.

* 1. **Scope:**

This project has following features and scopes:

* This College Voting System can be used for multiple colleges for voting process.
* College Voting System is a software application which maintains records of the students, Candidates, voting, etc.
* This system is designed for voting purpose which saves lot of time and counting is done within a single click.
* Colleges can register through online. Each college can create multiple voting process with the help of administrator.
* The system counts voting and displays voting result automatically.
  1. **Definitions, Acronyms and Abbreviations:**
* **SRS:**Software Requirements System
* **OS:**Operating System
* **PHP:**PHP: Hypertext Preprocessor
* **PC:** Personal Computer
* **MB:** Mega Bytes
* **RAM:**Random Access Memory
* **MySQL:**MySQL database Server
  1. **References:**
* www.w3schools.com
* www.tutorialspoint.com
* www.stackoverflow.com
* Web Database Applications with PHP and MySQL By Hugh E. Williams, ‎David Lane
* An integrated approach to Software Engineering By Pankaj Jalote
  1. **Overview:**

This SRS will allow for a complete understanding of what is to be expected of the “**College Voting System**” web application to be constructed. The clear understanding of the “**College Voting System**” and its functionality will allow for end users and will be used for the development of the future stages of the project.

1. **Overall description:**

This section of the SRS describes all general factors of the product and its requirements. This SRS explains detailed requirements of **College Voting System** website application.

* 1. **Product Perspective:**

Current developing project is the web application project developing using PHP. This system is developed using PHP and all the record stores in the MySQL database.

* 1. **Product Features**:

This product has following features:

* Mainly this project helps colleges for voting process. This project manages multiple college voting in a single system.
* The public users can view voting results through online. No registration required to view voting results.
* Only authenticated users can login to the system. The login id and password must be required to login the system.
* The student or college staff needs to login to the website to access all the features of the website.
* College staff can set or configure college election type, candidate details, election date, etc.
* Students can vote directly through online by selecting candidate.
* There are 3 types of users in this system: Admin, College staff and student.
* Administrator can view all features of the website.
  1. **User classes and characteristics:**

There are 3types of users using for this system.

* **Student:**

The students are voters who can register to the website by entering their profile details. After the verification the student can login and they can participate for voting.

* **Administrative staff:**

Administrative staffs are college management employees who can add college course details, semester details and college section.

* **Teaching staff:**

Teaching staffs can add election details and they can verify registered student records.

* **Administrator:**

Administrator is the owner of the website who can view or modify the complete features of the website.

* 1. **Design and implementation constraints:**
* The developed system should run under any platform (Unix, Linux, Mac, Windows etc.) that contains a web browser which supports PHP, JavaScript and AJAX.
* Internet connectivity is required to send mails.
* The user who is accessing the system should be authorized.
* MySQL database is the backend of the system. All the record stores in mysql database.
  1. **Assumptions and Dependencies:**
* The users should have basic knowledge of the computers. They must be trained well to handle the features provided by this system.
* Some of the details are required to be entered by the user and may not be generated automatically.
* Administrator is created in the system already.
* Roles and tasks are predefined.

1. **Specific requirements:**
   1. **External Interface Requirements:**

This chapter is an outline of the inputs and outputs of the project.

* + 1. **User Interfaces:**

Each part of the user interface intends to be as user friendly as possible. The fonts and buttons used will be intended to be very fast and easy to load on web pages. The pages will be kept light in space so that it won’t take a long time for the page to load.

* + 1. **Hardware Interfaces:**
* **Operating System:** Unix, Linux, Mac, Windows etc.
* **Processor:** Pentium or Higher.
* **RAM:** 312MB or Higher.
* 14”monitor
* Keyboard and mouse
  + 1. **Software Interfaces:**
* **Development tool:** PHP : Hypertext Preprocessor, JavaScript, Ajax
* **Scripting server:** Apache server
* **Data Base server:** MySQL
* **IDE:** Notepad++
  + 1. **Communication Interfaces:**

Internet connection must required.

* 1. **Functional Requirements:**
     1. **College module:**

In this module administrator can add college records.While adding college records this stores Administrative staff records. College staff can login to the website by entering login credentials. Administrative staffs can add teaching staff records, college records, semester records, etc. Teaching staffs can add election and voting details.

Sub modules of the module are:

* Add Administrative staff
* View administrative staff
  + 1. **College staff module**

In this module college staff can login to the website by entering login credentials. After the login the staff can login to the website by entering login credentials. There are two types of users using this system. They are Administrative staff and Teaching staff.

Sub modules of this module are:

* Administrative login panel
* Teaching staff login panel
* Update profile
* Change password
* Add teaching staff
* View teaching staff
  + 1. **College settings module**

In this module college administrators can add course details, semester details and section details which is available in the college.

Sub modules of this module are:

* Add course
* View course
* Add semester
* View semester
* Add section
* View section
  + 1. **Student account module:**

This module allows students to register by entering college, semester, section, roll number, name, contact number, address, etc. After the registration college administrative staff or teaching staff can verify registration details. After successful verification the students can login to system by entering login credentials.

Sub modules of this module are:

* Registration module
* Login module
* Student profile module
* Change password module
  + 1. **Voting schedule module:**

In this module college administrative staff or teaching staff can add or schedule voting for class representative, president election, etc.

Sub modules of this module are:

* Voting schedule
* View voting schedule
  + 1. **Candidate module:**

The college administrative staff or teaching staff can add candidate details for the voting by selecting existing student record. The candidate list displays during voting. Students can vote for registered candidate.

Sub modules of this module are:

* Add candidate
* View candidate
  + 1. **Voting module:**

In the voting module students can vote for candidates. Students can vote during voting schedule by selecting candidate. The student needs to login to the system for voting. The students can vote once for every election.

Sub modules of this module are:

* Voting Panel
* Vote module
  + 1. **Voting result module:**

This is the public page where everyone can view voting result, winners list, total votes received, etc. The system automatically calculates voting records.

Sub modules of this module are:

* Voting result module
  + 1. **Dashboard module:**

The dashboard module is for administrator. Administrator has full authority of the website. Admin can view election report of all colleges. Admin can add different types of voting records.

Sub modules of this module are:

* Add college
* View college
* Add voting type
* View voting types
* View election schedule
* View voting result
  1. **Document conventions:**

The format of this SRS is simple. Bold face and indentation is used on general topics and or specific points of interest. The remainder of the document will be written using the standard font Times New Roman.

* 1. **Intended users and Reading suggestions:**

This document is intended for software developers, document writers and for general discussions on the implementation decisions regarding the software.

* 1. **System features:**

The coding is done with following characteristics in mind:

* Ease of design to code translation
* Code efficiency
* Memory efficiency
* Response time
* Maintainability
* Security
* Simple ease to understand code
* Efficient and consistent logic

1. **Other non-functional Requirements:**
   1. **Performance Requirements:**

Performance requirements define acceptable response times for system functionality.

* The system is supposed to be having good memory space and RAM should be Above 256 MB preferably.
* The sound card and graphics card will have to be of good quality and capacity.
* The load time for user interface screens shall take no longer than three seconds.
* The log in information shall be verified within three seconds.
* Queries shall return results within three seconds.
  1. **Safety Requirements:**
* In case the students forget their password, they can recover the password in the Forgot Password panel
* The password stores in the database in the format of encrypted password.
  1. **Security Requirements:**
* Only authenticated users can access this system.
  1. **Software quality attributes:**
* **Reliability:**

This system is designed to have very simple database just to cater the exact need of “Online Voting”. It is tested for all the constraints at development stage.

* **Availability:**

This system will only available till the system on which it is installed is running.

* **Security:**

This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use the application. If the legitimate users share the authentication information then the system is open to outsiders.

* **Maintainability:**

There is maintenance required for the website. The database is provided by the Administrator as well as the end-use.

* **Portability:**

The system works anywhere with the internet connection.

1. **Other requirements:**

None

SYSTEM DESIGN

**1. Introduction:**

The purpose of the designing phase is to plan solution to the problem specified by the required document. This phase moves from the problem domain to the solution domain. The design activity often results in three separate outputs:

1. Architecture Design
2. High Level Design
3. Detailed Design

In architecture design the focus is on identifying components or subsystem and how they interact with each other. The high level design identifies the module that should be built for developing the system. In case of detailed design, the focus is on how the modules are implemented in software.

System design is a solution, a “How to” approach to the creation of a new system. The important phase is composed of several steps. It provides the understanding of procedural details necessary for implementing the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specification.

**2.Context Flow Diagram:**

The System Context Design or Context Flow Diagram (CFD) describes the external entities acting on the system. The environment in which the system is used is depicted in the figure.

**2.1 Context Flow Diagram**

****

**2.2 Data Flow Diagram:**

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of the data through an information system. A DFD also can be used for the visualization of data processing. It is a common practise for a designer to draw a context level DFD first which shows the interaction between the system and outside entities. The context level DFD is the exploded to show more detail of the system being modelled.

Data Flow Diagram describes what data flow (logical) rather than how they are processed, so it depend on hardware, software, data structure or file organisation.

The DFD uses four symbols, and are explained below:

* A **rectangle,** which defines the source or destination of system data also called as external entity. An external entity is not responsible for any task performed by the system.
* An **arrow** represents data flow. It represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities. It must be given a name above the arrowhead showing the direction of flow.

* A **circle** or a **Bubble** represents a process that transforms data from one form to another by performing some tasks with the data. The process name must be given a general idea of its function.
* Two **horizontal parallel lines** represents data store, a data store is place where data is held temporarily from one transaction to the next or is permanently.

**Top Level Data Flow Diagram**

****

**DFD Level 3.3.1**



**Input:**

* College Details
* Administrative Staff Details
* Course details
* Semester record
* Section details

**Process:**

* Adding College
* Adding Administrative Staff
* Adding Course Details
* Adding semester records
* Adding Sections

**Output:**

* College Details are inserted in college table.
* Administrative staff details are inserted in staff table.
* Course details are added in course table.
* Semester details are added in semester table.
* Section details are added in section table.

**DFD Level 3.3.2**

****

**Input:**

* Election types
* Teaching staff profile
* Login details
* Election details

**Process:**

* Adding election types
* Adding teaching staff profile
* Loading course ,semester details
* Scheduling election

**Output:**

* Election type details are added in electiontype table.
* Teaching staff profiles are added in staff table.
* Courses are added in course table and Semester details are added in semester table.
* Election settings like (time,year,date) are added in election table.

**DFD Level 3.3.3**

****

**Input:**

* Student registration details
* Student verification details
* Exam result details
* Candidate details

**Process:**

* Adding student registration details
* Student registration is verified
* Student applying as candidate

**Output:**

* Student registration details are stored in student table
* Student registration is verified by teaching staff
* Teaching staff updates the exam\_result and it is stored in exam\_result table.
* Candidate details are stored in candidate table.

**DFD Level 3.3.4**

****

**Input:**

* Login credentials
* Viewing voting result
* Administrative staff

**Process:**

* Login process
* Voting process
* Publishing voting\_result

**Output:**

* Student login details is stored in student table.
* Election details are stored in election table
* Administrative staff publishes voting result and results are stored in voting\_result table.

DATABASE DESIGN

DATABASE DESIGN DOCUMENT

**1. Introduction:**

Database Design maintains the data required by the system. One of the key design issues involved in the database to design is the distribution of data in a way that minimizes transaction traffic. Another key design issue is the choice of the database management system. Database Tables used are described in the following sections.

**Database:**

A database is a collection of data organized to allow easy access of retrievals, additions, modifications and deletions.

A typical database consists of different parts as shown in the following figure. The Database server (MySQL Server) in the top level is a software that can be accessed by multiple users. Within the database server, number of database can be stored.

Each database stored data in the series of tables that can be related to each other in different ways. The most widely used approach for structuring the data is called Relational database system (RDBMS).A RDBMS is a collection of tables of data.

Database design is a process of organizing the data in an orderly manner so as to provided easy to the required information.

It is difficult to Maintain the database if lot of repetitive data stored in the table. If one instance of the data undergoes a change, that undergoes a change has to be made for all occurrences of the data. To eliminate duplication and easy maintenance of data, it is recommended to create a table of repeated values in one table called master table. The reference remains same throughout the other tables.

**4.2 NORMALIZED TABLES:**

**Table structure for table candidate**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***candidateid*** | int(10) | Primary key | Candidate ID |
| electionid | int(10) | Not null | Election ID |
| studentid | int(10) | Not null | Student ID |
| status | varchar(10) | Not null | Candidate status |

**Table structure for table College**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***collegeid*** | int(10) | Primary key | College ID |
| collegename | varchar(50) | Not null | College Name |
| collegelogo | varchar(100) | Not null | College Logo |
| collegeaddress | Text | Not null | College Address |
| contactno | varchar(15) | Not null | Contact Number |
| status | varchar(10) | Not null | College Status |

**Table structure for table course**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***courseid*** | int(10) | Primary key | Course ID |
| course | varchar(25) | Not null | Course |
| description | text | Not null | Course Description |
| status | varchar(10) | Not null | Course Status |
| collegeid | int(10) | Foreign key | College ID |

**Table structure for table election**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***electionid*** | int(10) | Primary key | Election ID |
| electiontypeid | int(10) | Foreign key | Election Type ID |
| electionyear | int(100) | Not null | Election Year |
| electiondate | date | Not null | Election Date |
| startdttime | datetime | Not null | Start Date Time |
| enddttime | datetime | Not null | End Date Time |
| courseid | int(10) | Foreign key | Course ID |
| semid | int(10) | Foreign key | Semester ID |
| sectionid | int(10) | Foreign key | Section ID |
| Icon | varchar(100) | Not null | Icon |
| description | text | Not null | Election Description |
| status | varchar(10) | Not null | Election Status |

**Table structure for table electiontype**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***electiontypeid*** | int(10) | Primary key | Election Type ID |
| electiontype | varchar(30) | Not null | Election Type |
| description | text | Not null | Election Type Description |
| status | varchar(10) | Not null | Election Type Status |
| election\_for | varchar(20) | Not null | Election For |
| collegeid | int(10) | Foreign key | College ID |

**Table structure for table exam\_result**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***exam\_resultid*** | int(10) | Primary key | Exam Result ID |
| studentid | int(10) | Foreign key | Student ID |
| courseid | int(10) | Foreign key | Course ID |
| semid | int(10) | Foreign key | Semester ID |
| result | varchar(25) | Not null | Result |
| status | varchar(10) | Not null | Exam Result Status |

**Table structure for table section**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***sectionid*** | int(10) | Primary key | Section ID |
| semid | int(10) | Foreign key | Semester ID |
| section | varchar(25) | Not null | Section ID |
| description | text | Not null | Section Description |
| status | varchar(10) | Not null | Section Status |

**Table structure for table semester**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***semid*** | int(10) | Primary key | Semester ID |
| courseid | int(10) | Foreign key | Course ID |
| semester | int(10) | Foreign key | Semester |
| description | text | Not null | Semester Description |
| status | varchar(10) | Not null | Semester Status |

**Table structure for table staff**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***staffid*** | int(10) | Primary key | Staff ID |
| stafftype | varchar(35) | Not null | Staff Type |
| staffname | varchar(50) | Not null | Staff Name |
| loginid | varchar(25) | Not null | Login ID |
| password | varchar(100) | Not null | Password |
| status | varchar(10) | Not null | Staff Status |
| collegeid | int(10) | Foreign key | College Id |

**Table structure for table student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***studentid*** | int(10) | Primary key | Student ID |
| courseid | int(10) | Foreign key | Course ID |
| semid | int(10) | Foreign key | Semester ID |
| sectionid | int(10) | Foreign key | Section ID |
| studentname | varchar(25) | Not null | Student Name |
| rollno | varchar(20) | Not null | Roll Number |
| password | varchar(100) | Not null | Password |
| studentimg | varchar(100) | Not null | Student Image |
| description | text | Not null | Student Description |
| status | varchar(10) | Not null | Student Status |
| collegeid | int(10) | Foreign key | College Id |

**Table structure for table voting**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***votingid*** | int(10) | Primary key | Voting ID |
| electionid | int(10) | Foreign key | Election ID |
| studentid | int(10) | Foreign key | Student ID |
| candidateid | int(10) | Foreign key | Candidate ID |
| votindttime | datetime | Not null | Voting Date Time |
| status | varchar(10) | Not null | Voting Status |

**Table structure for table voting\_result**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Index** | **Description** |
| ***voting\_resultid*** | int(10) | Primary key | Voting Result ID |
| Electionid | int(10) | Foreign key | Election ID |
| Candidateid | int(10) | Foreign key | Candidate ID |
| Noofvoting | int(10) | Not null | Number Of Voting |
| Status | varchar(10) | Not null | Voting Result Status |

4.3 **Entity-Relationship Diagram:**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the [relationships between entities in a database](http://databases.about.com/od/specificproducts/a/Database-Relationships-An-Introduction-To-Foreign-Keys-Joins-And-E-R-Diagrams.htm). ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

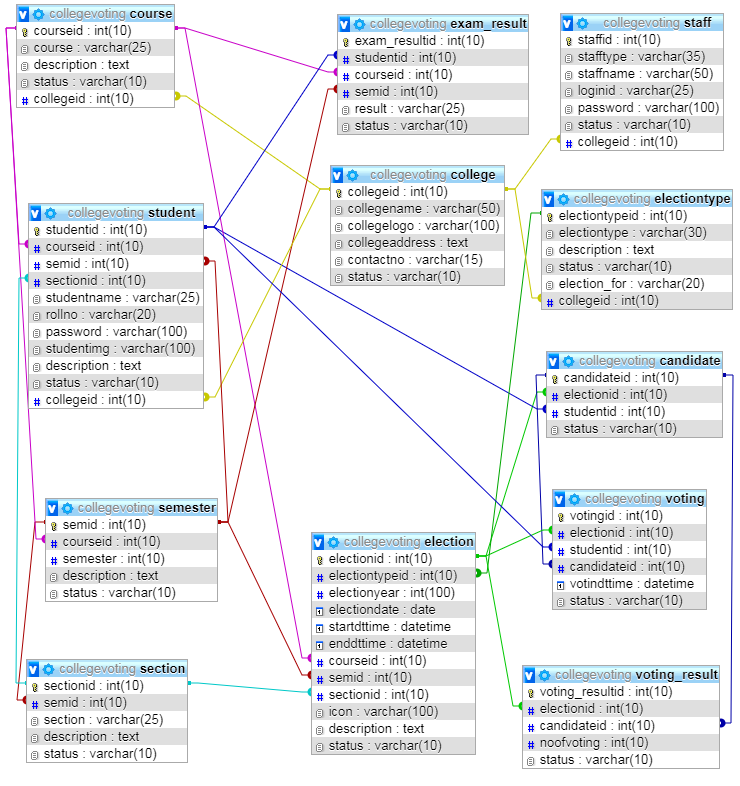
The Symbols are shown in below table:

|  |  |  |
| --- | --- | --- |
| **Name** | **Notation** | **Description** |
| Entity | https://lh6.googleusercontent.com/155nI7xT03eaI9Q2Akt1rW7al3lClN5J6wS0tgCoeEjd1tfXTCPjSJgkNujMs9k0mOH7SJxFlUwXD0BnlXR1yhyZ53QWQ6ukyH3ATmrnhaVnw9LA0OHN1Et1w_3i2bS6q5W4rQgj7PJ9-gBY5g | Entity is represented by a box within the ERD.  Entities are abstract concepts, each representing one or more instances of the concept in question. An entity might be considered a container that holds all of the instances of a particular thing in a system.  Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity. |
| Relationship | https://lh6.googleusercontent.com/zUC9tciH6o2PuvLczLcQ1ZnYyhEFBr1f-TeIQ4TmDeh4qgFrrrsKkK75o9l3iLSIBzpE7A45u7iiOATOEhiiU0fbSibYzsLCCkbOvEHuuLHT4zRkS_kjRcTv6HNjqFslCEm4jyIU5aC2EXzwLg | Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the objects. |
| Attributes | https://lh6.googleusercontent.com/YJGUrVZNxyDrGc7nvdIxfh5MWXc7MRcyinoxxMOVI_FsVBcf2PNjVT1vIchH7MYhD3LVtQM7vVZzzCExUfcw7qo2mfwpOO8wtswPvoY0F8Ugist3afnZKNb8GOoyoEExdwt7ehnmVDuGN1Gi6Q | Attributes are represented by Oval. An attribute is a single data item related to a database object. The database schema associates one or more attributes with each database entity. |

**ER-Diagram :**



**Database Schema Diagram:**



detailed design

           Detailed Design

1. **Introduction :**

This project is aimed at developing a Voting System for Colleges. The main objective of developing this system is for voting purpose which saves lot of time in counting process. It will make voting process fully computerized which is very fast and efficient. Also this web application maintains college students record, candidates record and voting record. This software can be used by any colleges to make election computerized.

College Voting System offers many advantages over the conventional paper-based electoral systems, including mobility and convenience for voters, greater speed and accuracy in the counting process, prevention of involuntary voting errors, better accessibility, lower costs, support of multiple languages, greater flexibility, etc. It has several core functionalities: security, ease-of-use, accessibility, and scalability/flexibility.

1. **Applicable documents :**

     The documents used during detailed design are

* System   Requirements Document
* System Design
* Database Design

1. **Structure of the software package** :

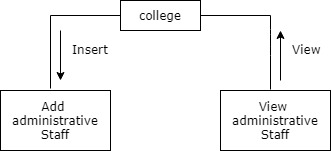
           The Components are as Follows –

* Component for College
* Component for Staff
* Component for College Settings
* Component for Student Account
* Component for Voting Schedule
* Component for Voting
* Component for Dashboard

1. **Modular Decomposition Of Components :**

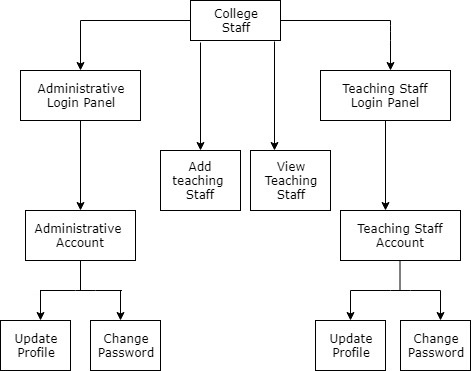
**4.1. College component :**

**4.1.1. Structure chart for college :**



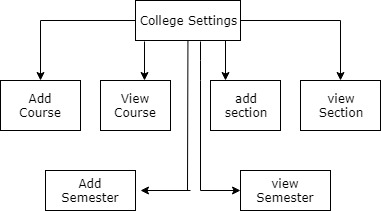
**4.2. College staff Component :**

**4.2.1. Structure chart for college staff :**



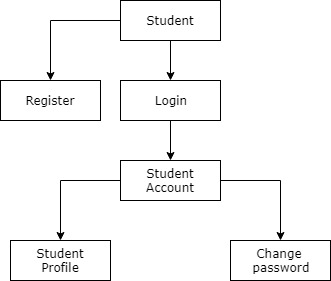
**4.3. College settings Component :**

**4.3.1. Structure chart for College settings :**



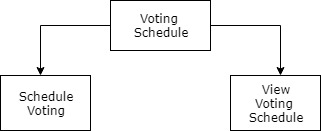
**4.4. Student Account Component :**

**4.4.1. Structure chart for Student Account:**



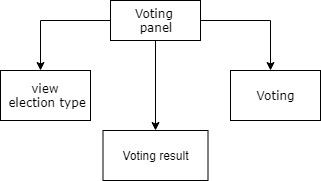
**4.5. Voting Schedule Component :**

**4.5.1. Structure chart for Voting Schedule:**



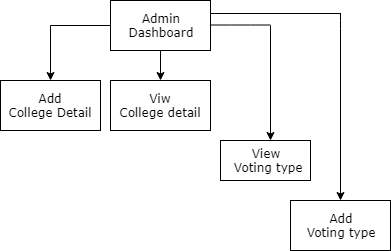
**4.6. Voting Component :**

**4.6.1. Structure chart for Voting :**



**4.7. Dashboard Component :**

**4.7.1. Structure chart for dashboard :**



Testing

And

Implementation

Testing Overview

**7.1 Introduction**

Software testing is an investigation conducted to provide stack holders with information about the quality of the product or service under test. Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is oriented to ‘detection’.

Software testing has three main purposes:

* The verification process confirms that the software meet its technical specifications. A “specification” is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
* The validation process confirms that the software meets the business requirements.
* A defect is a variance between the expected and actual result. The defect’s ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

There are two types of software testing:

* Black box testing-internal system design is not considered in this type of testing. Tests are based on requirements and functionality.
* White box testing-this testing is based on knowledge of thaw internal logic of an application’s code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as ‘for condition x your derived result is y’; where as other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software testing process.

**7.2 Testing objectives**

* Finding defects which may get created by the programmer while developing the software
* Gaining confidence in and providing information about the level of quality
* To prevent defects
* To make sure that the end results meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification.

**7.3 Testing methods**

System testing is the stage of implementation. This is to check whether system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: online response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

**7.4 Testing steps:**

**7.4.1 Unit testing**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

**7.4.2 Integration testing**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

**7.4.3 Validation**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

* The function or performance characteristics confirm to specification and are accepted.
* A deviation from specification is uncovered and a deficiency lists is created.
* Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

**7.4.4 Output testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

**7.4.5 User acceptance testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

* Input screen design.
* Output screen design.
* Online message should be guide to the user.
* Format of reports and other outputs.

**7.5. Test cases**

* **Admin Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If field in the form empty. | Value of form fields. | Invalid login credentials entered. | SUCCESSFUL |
| 2. | If login id and password does not match | Login id,Password | Invalid login credentials entered. | SUCCESSFUL |

* **Admin Profile**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If field in the form empty. | Value of form fields. | Alert the user to enter the fields and proceed | SUCCESSFUL |
| 2. | If staff name is empty | Staff Name | Kindly enter staff name | SUCCESSFUL |
| 3. | If staff name is other than alphabets | Staff Name | Staff Name is not valid | SUCCESSFUL |
| 4. | If login id is empty | Login id | Please enter the loginid | SUCCESSFUL |

* **Change password**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If the fields in the form is empty. | Value of form fields. | Alert the user to enter the fields and then proceed | SUCCESSFUL |
| 2. | If the old password is empty. | Old password | Kindly enter the old password. | SUCCESSFUL |
| 3. | If the new password is empty. | New password | Kindly enter the new password. | SUCCESSFUL |
| 4. | If the new password is not between the given range. | New password | New password must contain more than 6 characters. | SUCCESSFUL |
| 5. | If the confirm password is empty. | Confirm password | Kindly enter the Confirm Password. | SUCCESSFUL |
| 6. | If the confirm password and new password does not match | Confirm password,New password | Entered password and confirm password is not matching. | SUCCESSFUL |

* **Adding election type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If the fields in form is empty. | Value of form fields | Alert the user to enter the fields. | SUCCESSFUL |
| 2. | If college is empty. | College | Kindly select the college. | SUCCESSFUL |
| 3. | If election type is empty. | Election type | Kindly set the election type. | SUCCESSFUL |
| 4. | If election for is empty. | Election for | Kindly select the election for | SUCCESSFUL |
| 5. | If status is empty. | Status | Kindly select the status | SUCCESSFUL |

* **Add new election**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If the fields in the form is empty. | Value of form fields. | Alert the user to enter the fields. | SUCCESSFUL |
| 2. | If election type is empty. | Election type | Kindly select the election type. | SUCCESSFUL |
| 3. | If election date is empty. | Election date | Kindly select the election date. | SUCCESSFUL |
| 4. | If the election start time is not set. | Start time | Kindly select the start time. | SUCCESSFUL |
| 5. | If the election end time is not set. | End time | Kindly select the end time. | SUCCESSFUL |
| 6. | If the status is empty. | Status | Kindly select the status. | SUCCESSFUL |

* **Publishing of exam result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If course is empty. | Course | Kindly select the course. | SUCCESSFUL |
| 2. | If semester is empty. | semester | Kindly select the semester. | SUCCESSFUL |
| 3. | If section is empty. | Section | Kindly select the section | SUCCESSFUL |

* **Student registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If fields in the form is empty. | Value of form fields. | Alert the user to enter the fields. | SUCCESSFUL |
| 2. | If college is empty | College | Kindly select the college | SUCCESSFUL |
| 3. | If course is empty. | Course | Kindly select the course. | SUCCESSFUL |
| 4. | If semester is empty. | Semester | Kindly select the semester. | SUCCESSFUL |
| 5. | If section is empty. | section | Kindly select the section. | SUCCESSFUL |
| 6. | If student name is empty. | Student name | Kindly enter the student name. | SUCCESSFUL |
| 7. | If student name is other than alphabets. | Student name | Student name is not valid. | SUCCESSFUL |
| 8. | If roll number is empty | Roll no | Kindly enter roll number | SUCCESSFUL |
| 9. | If roll number is other than numbers. | Roll no | Invalid roll number | SUCCESSFUL |
| 10. | If password is empty. | Password | Kindly enter the password. | SUCCESSFUL |
| 11. | If the password is not between the given range | password | Password should contain more than six characters. | SUCCESSFUL |
| 12. | If the confirm password is empty. | Confirm password | Kindly enter the confirm password. | SUCCESSFUL |
| 13. | If the password and confirm password is not matching. | Password,Confirm password | Entered password and confirm password not matching. | SUCCESSFUL |
| 14. | If status is not selected. | Status | Kindly select the status. | SUCCESSFUL |

* **Adding college**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If fields in the form is empty. | Value of form fields. | Alert the user to enter the fields. | SUCCESSFUL |
| 2. | If college is empty. | College | Kindly enter the college name | SUCCESSFUL |
| 3. | If college address is empty | College address | Kindly enter the college address. | SUCCESSFUL |
| 4. | If contact number is empty | Contact number | Kindly enter contactno | SUCCESSFUL |
| 5. | If contact number other than digits. | Contact number | Contact number is not valid. | SUCCESSFUL |
| 6. | If contact number is not 10 digits. | Contact number | Contact number should contain 10 digits. | SUCCESSFUL |
| 7. | If status is not selected. | Status | Kindly select the status. | SUCCESSFUL |

* **Adding staff records**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If college is empty | College | Kindly select the college. | SUCCESSFUL |
| 2. | If staff type is not selected. | Staff type | Kindly select the staff type. | SUCCESSFUL |
| 3. | If staff name is empty. | Staff name | Kindly enter the staff name. | SUCCESSFUL |
| 4. | If login id is empty. | Login id | Please enter the loginid | SUCCESSFUL |
| 5. | If password is empty. | Password | Kindly enter password. | SUCCESSFUL |
| 6. | If password is not in the given range. | Password | Password should contain more than six characters. | SUCCESSFUL |
| 7. | If confirm password is empty. | Confirm password | Kindly enter confirm password | SUCCESSFUL |
| 8. | If password and confirm password does not match. | Password,confirm password | Entered password and confirm password not matching. | SUCCESSFUL |
| 9. | If status is not selected | Status | Kindly select status. | SUCCESSFUL |

* **Student login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If fields in the form is empty. | Value of form fields. | Alert the user to enter the fields. | SUCCESSFUL |
| 2. | If the roll is empty. | Roll number | Kindly enter the rollno. | SUCCESSFUL |
| 3. | If password is empty. | password | Kindly enter the password. | SUCCESSFUL |
| 4. | If rollno and password is not matching. | Rollno,password | Invalid login credentials entered. | SUCCESSFUL |

* **Adding teaching staff by administrative staff**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If staff name is empty. | Staff name | Kindly enter staff name | SUCCESSFUL |
| 2. | If loginid is empty. | Login id | Kindly enter login id | SUCCESSFUL |
| 3. | If password is empty. | password | Kindly enter password. | SUCCESSFUL |
| 4. | If password is not in the given range. | password | Password should contain more than six characters. | SUCCESSFUL |
| 5. | If the confirm password is empty. | Confirm password | Kindly enter confirm password | SUCCESSFUL |
| 6. | If password and confirm password does not match | Password,confirm password | Entered password and confirm password not matching. | SUCCESSFUL |
| 7. | If status is not selected. | Status | Kindly select the status. | SUCCESSFUL |

* **Adding course**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If course is empty. | Course | Kindly add the course. | SUCCESSFUL |
| 2. | If status is empty. | Status | Kindly select the status. | SUCCESSFUL |

* **Adding semester**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If course is empty. | Course | Kindly add the course. | SUCCESSFUL |
| 2. | If semester is empty. | Semester | Kindly enter the semester. | SUCCESSFUL |
| 3. | If status is empty. | Status | Kindly select the status. | SUCCESSFUL |

* **Adding section**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition to be tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If course is empty. | Course | Kindly add the course. | SUCCESSFUL |
| 2. | If semester is empty. | Semester | Kindly enter the semester. | SUCCESSFUL |
| 3. | If section is empty | Section | Kindly enter the section | SUCCESSFUL |
| 4. | If status is empty. | Status | Kindly select the status. | SUCCESSFUL |

**7.6 Implementation**

Implementation is the stage of the project when the theoretical is turned into a working system. At this stage the workload and the latest upheaval shifts to the user departments. If the implementation stage is not clearly planned and controlled it can cause choice. The term implementation has different meanings, ranging from the conversion of the basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised system design into an operational one. During the implementation stage we convert the detailed code in a programming language.

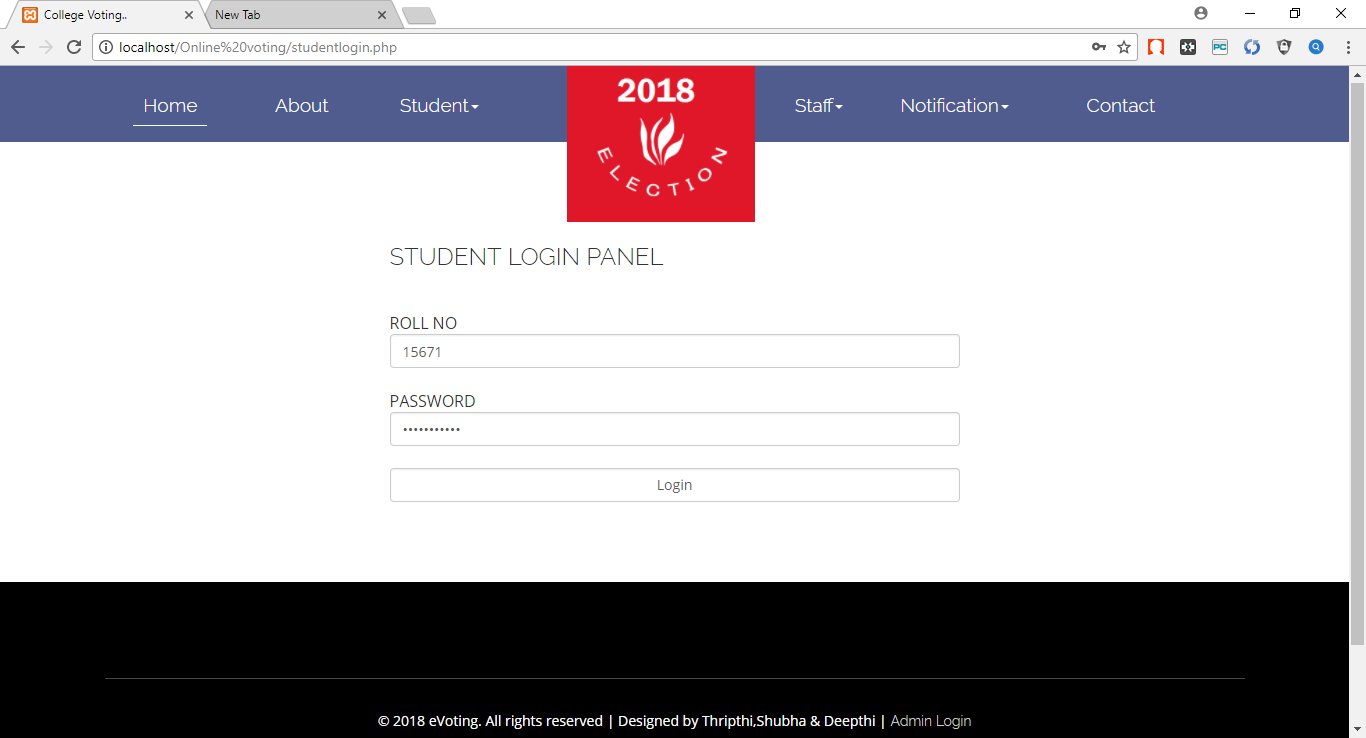
User manual

USER MANUAL

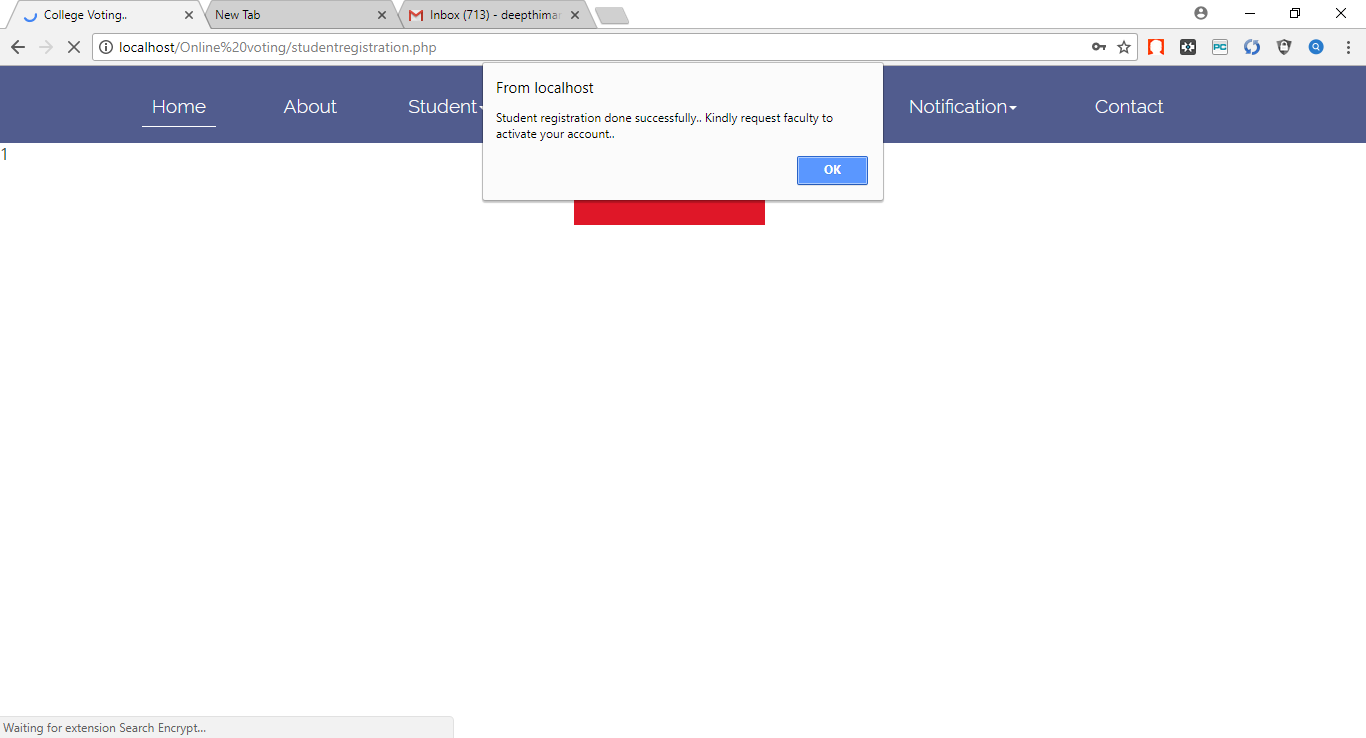
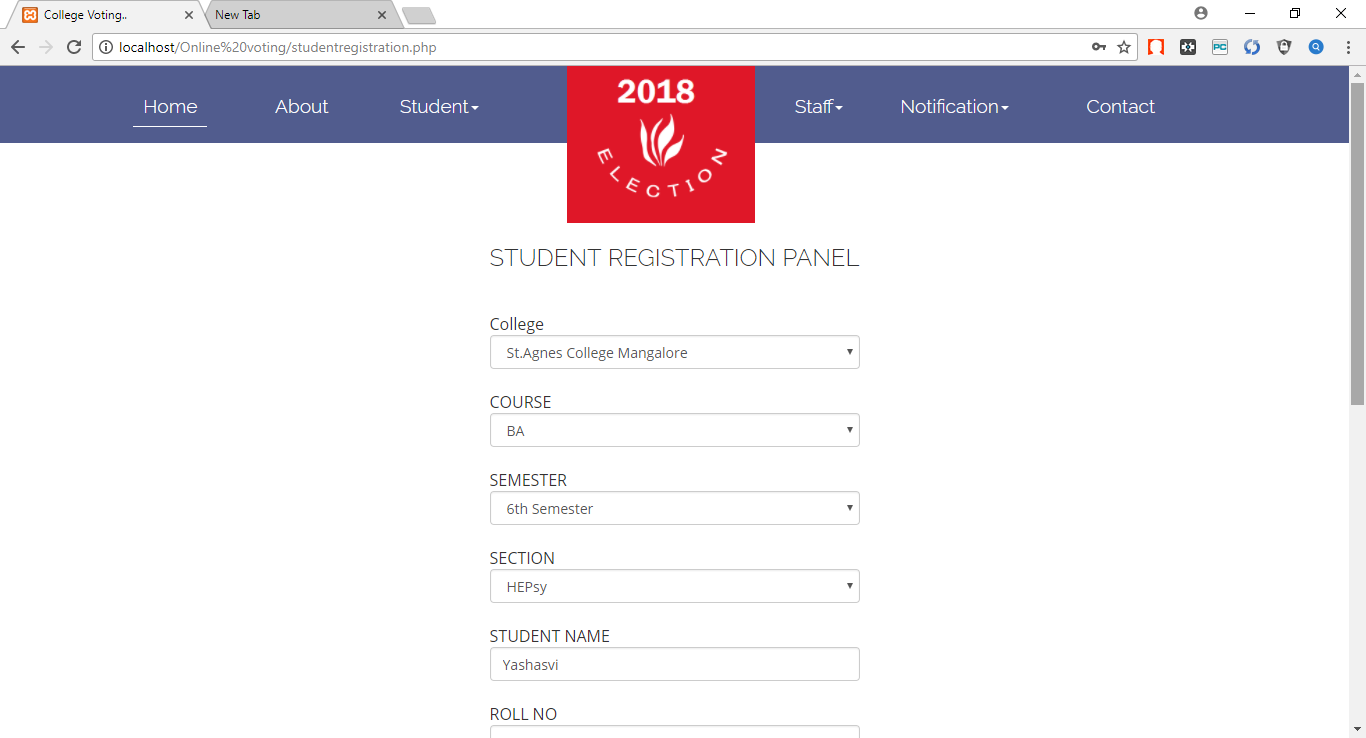
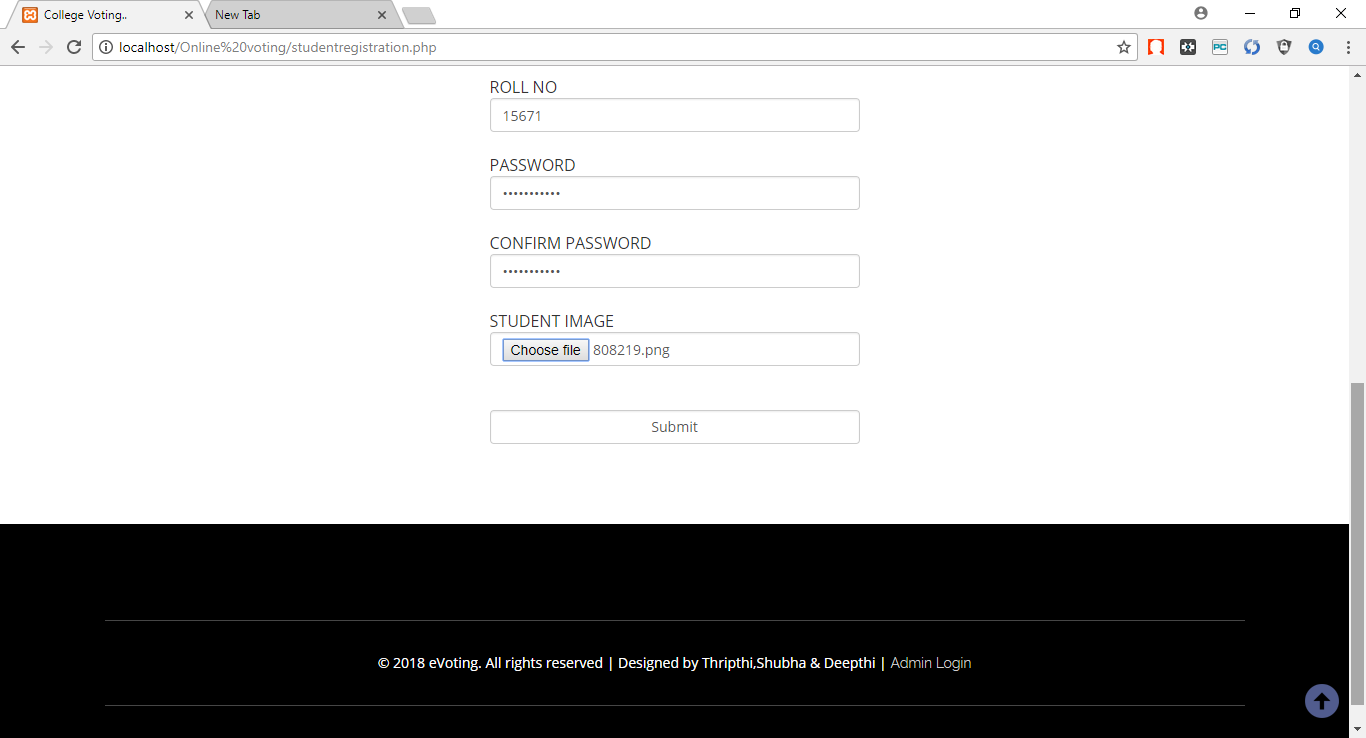
User manual gives the rough idea about the working of the system,so that any user can easily interact with the system. It includes entire details starting from how one can login to the system; it tells what can be done after logging into the system. User manual helps the new user to interact with the system.

College voting system is a user friendly web application which is developed using PHP and Ajax as front end and MySql as the back end. In order to run this software the minimum software requirements are the system should be expected under windows operating system.

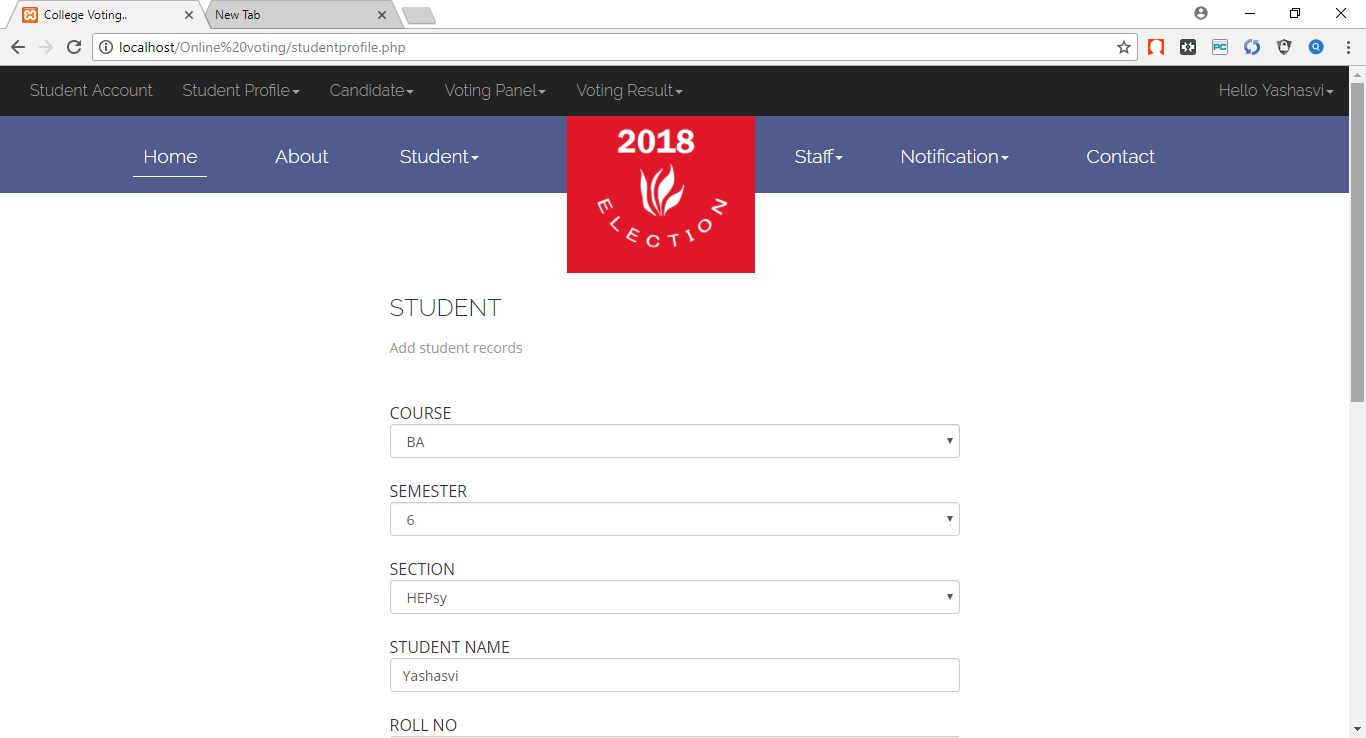
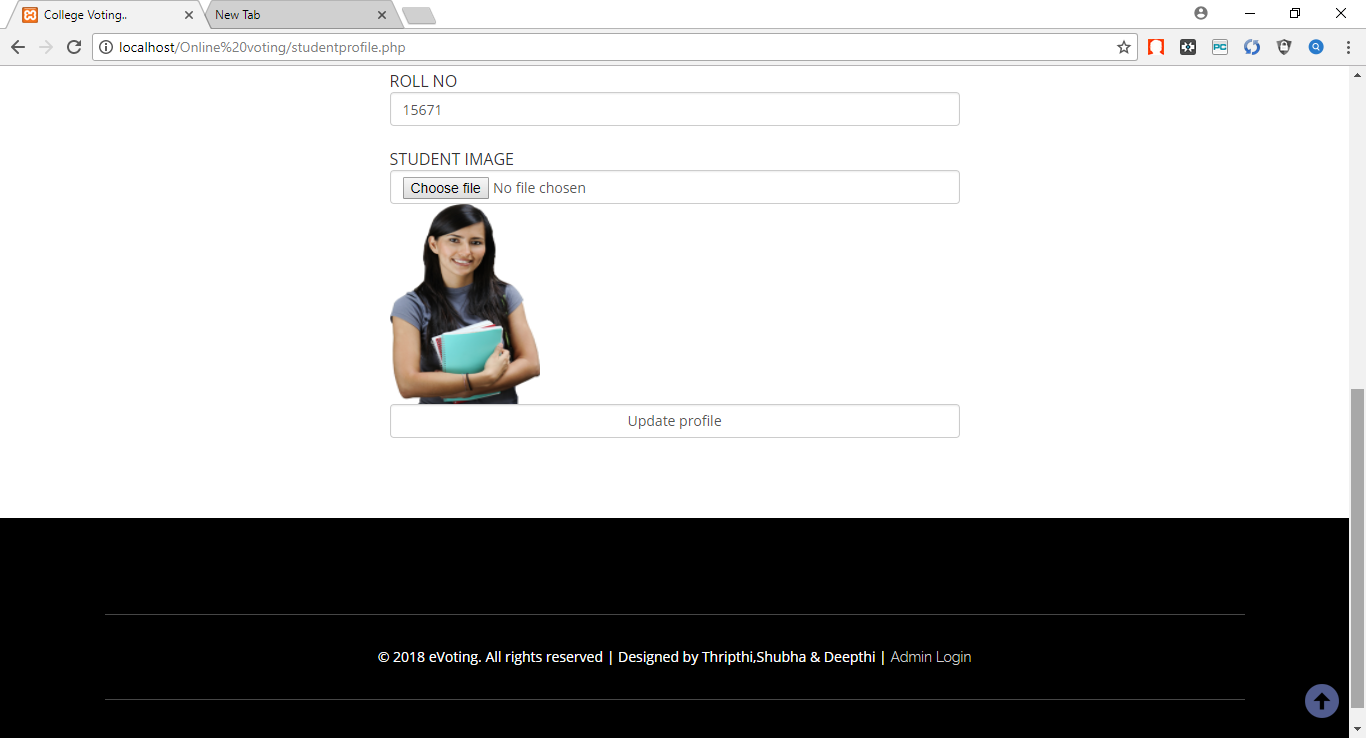
* **Home Page:**

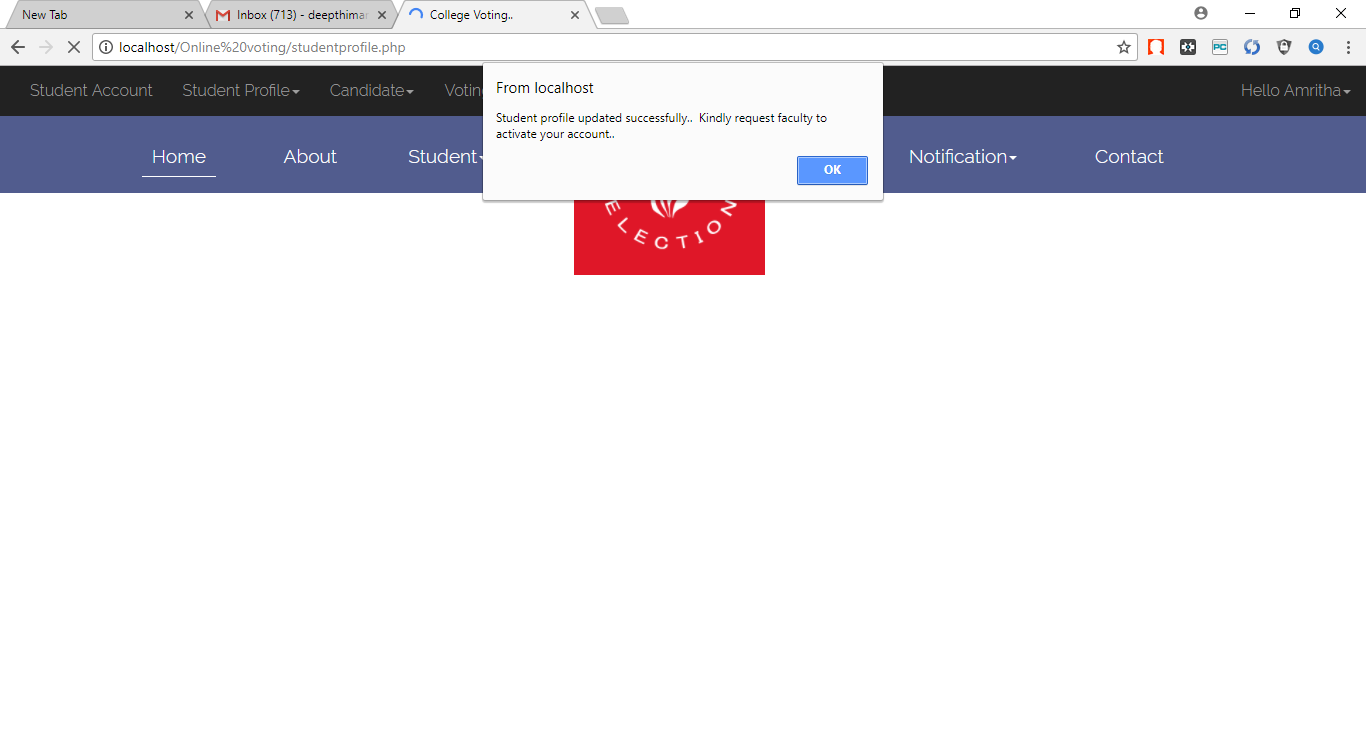


**This is the main page of college voting site. Here the student decides whether to login to the website or not.**

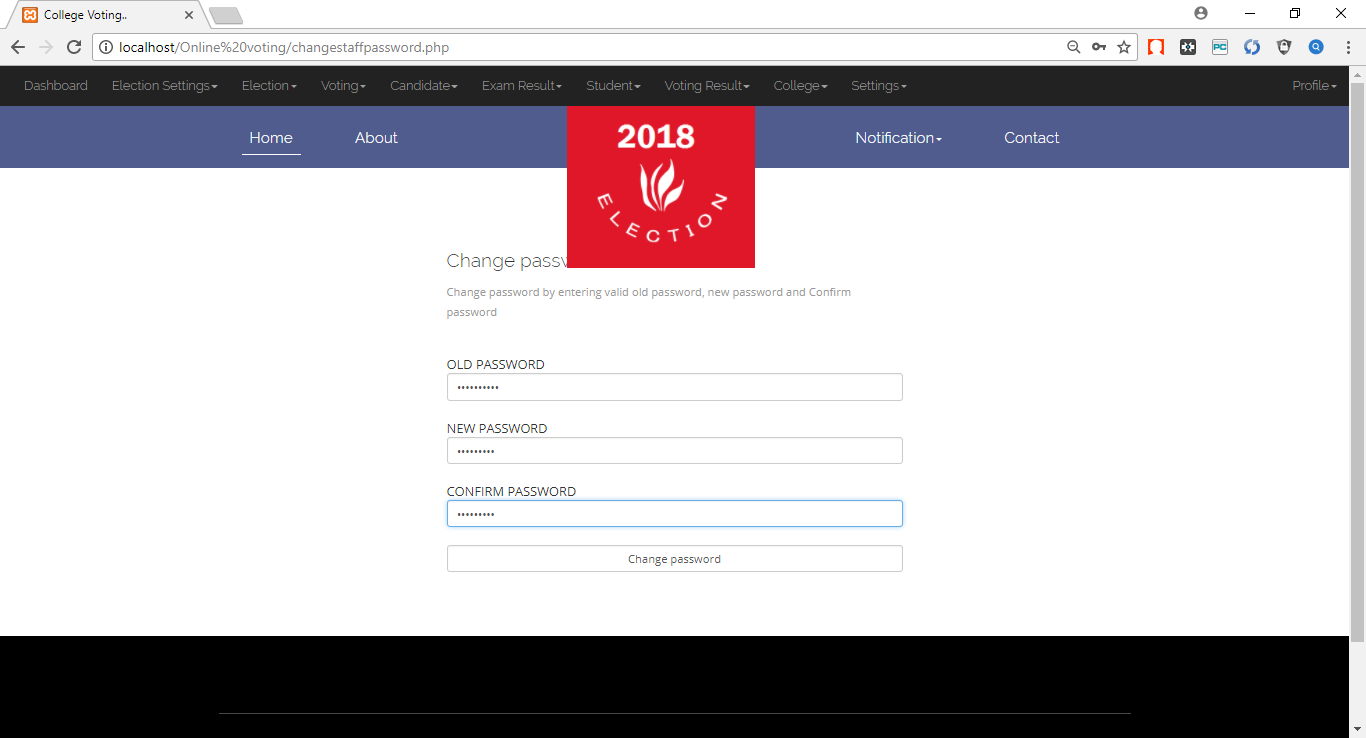
* **To login as a student click on the login button after providing correct LoginID and Password.**
* **Click on the Admin Login button to go to the Admin Login Panel.**
* **Registration Panel Page:**

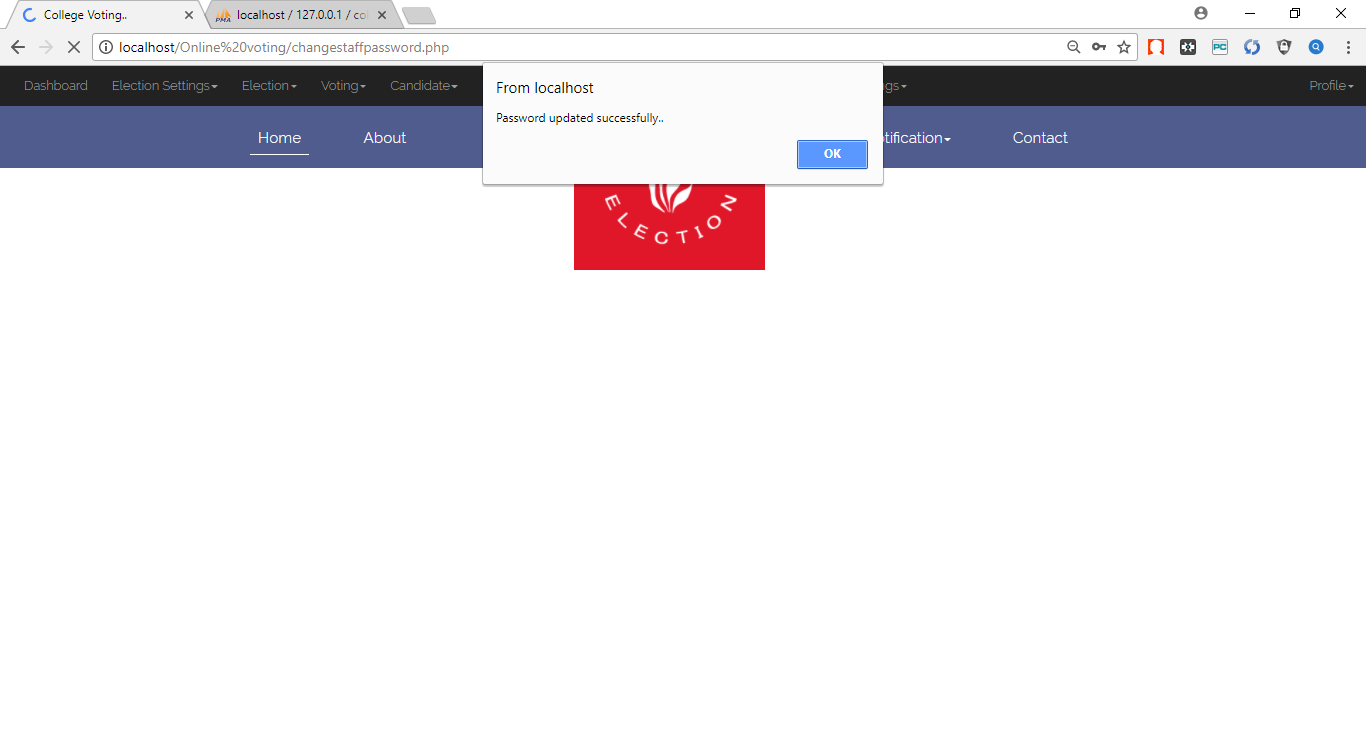
New student registration takes place using this panel.

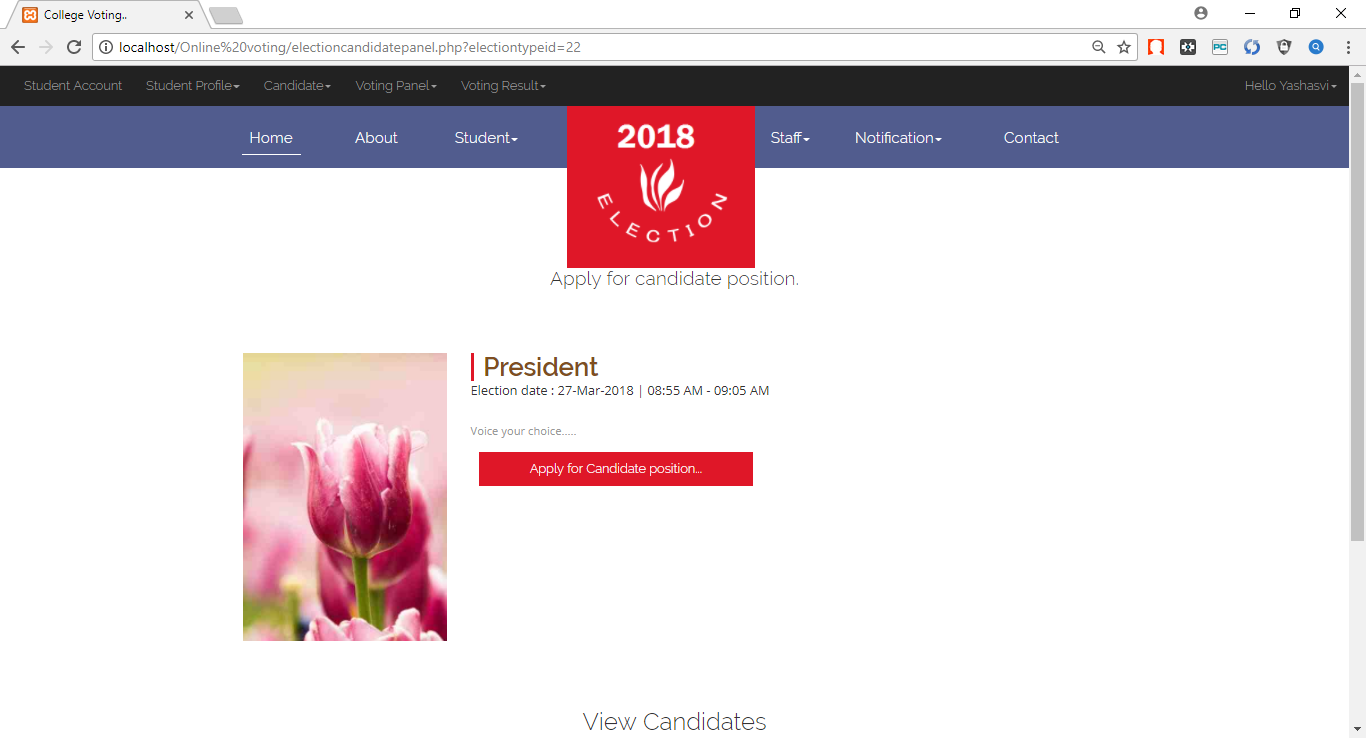
* **Student profile:**

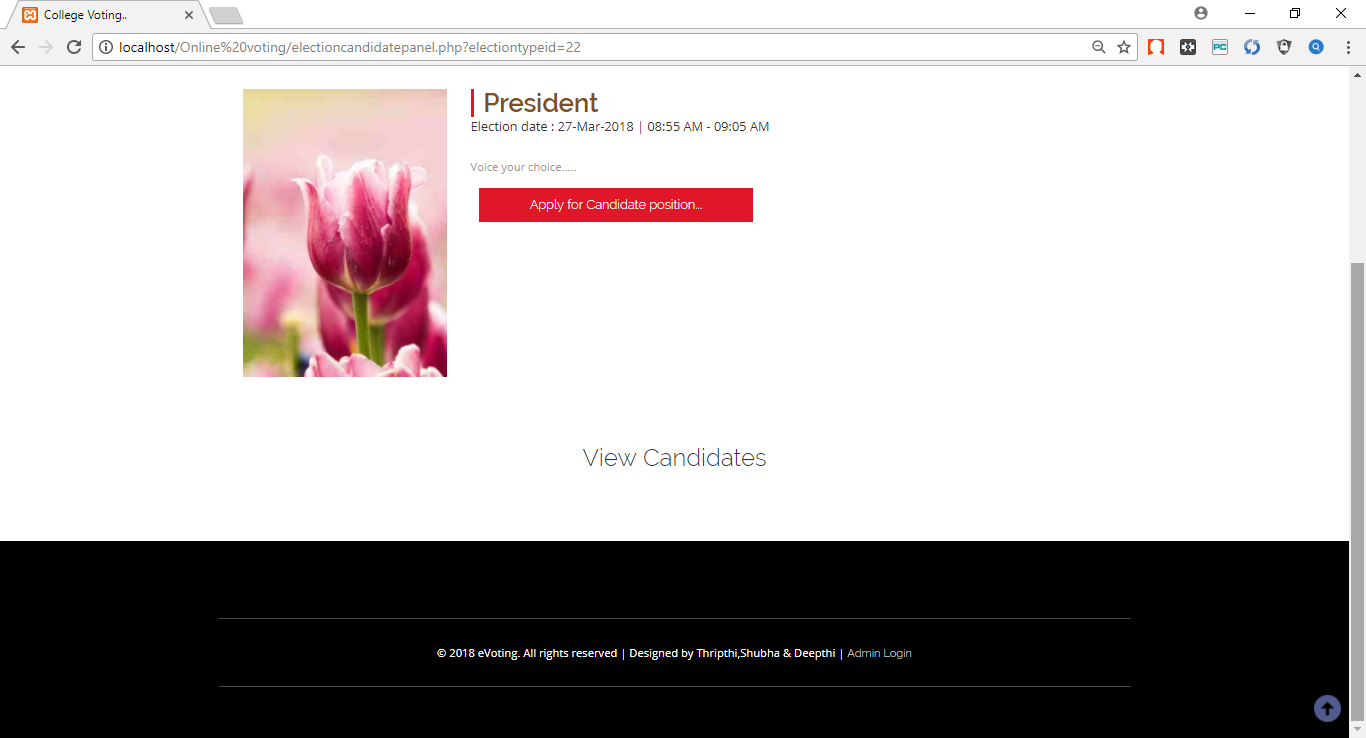


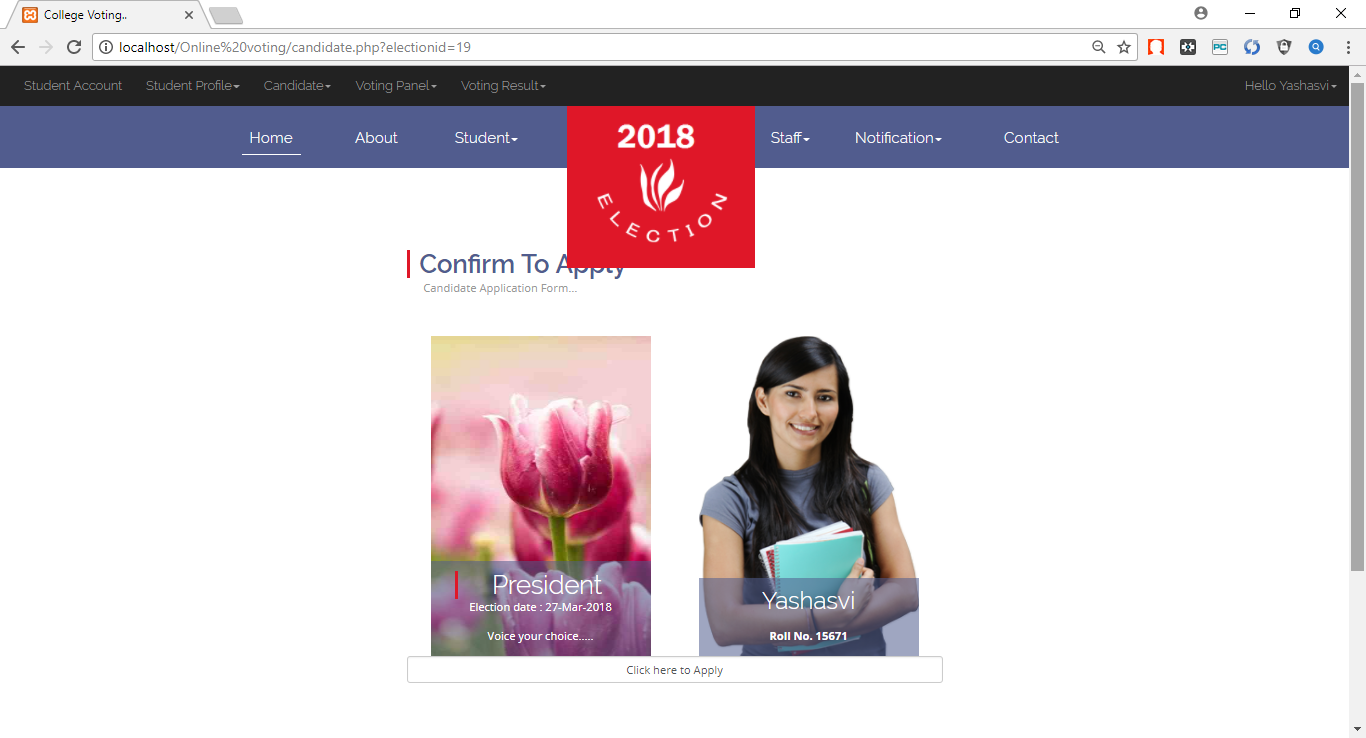
It is the student profile page to view her personal details. Here student can change their name and loginid. But they have to wait for the approval from the faculty.

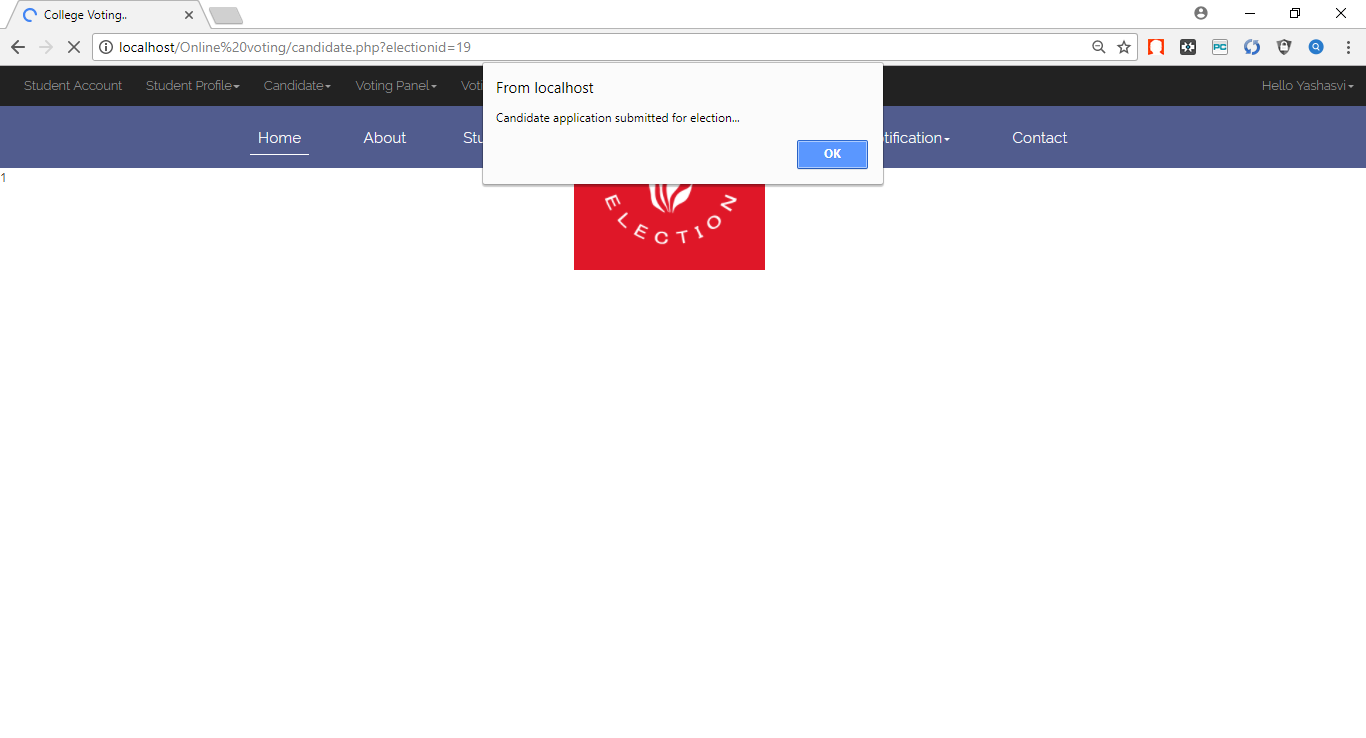
* **Change login password**



* **Candidate**

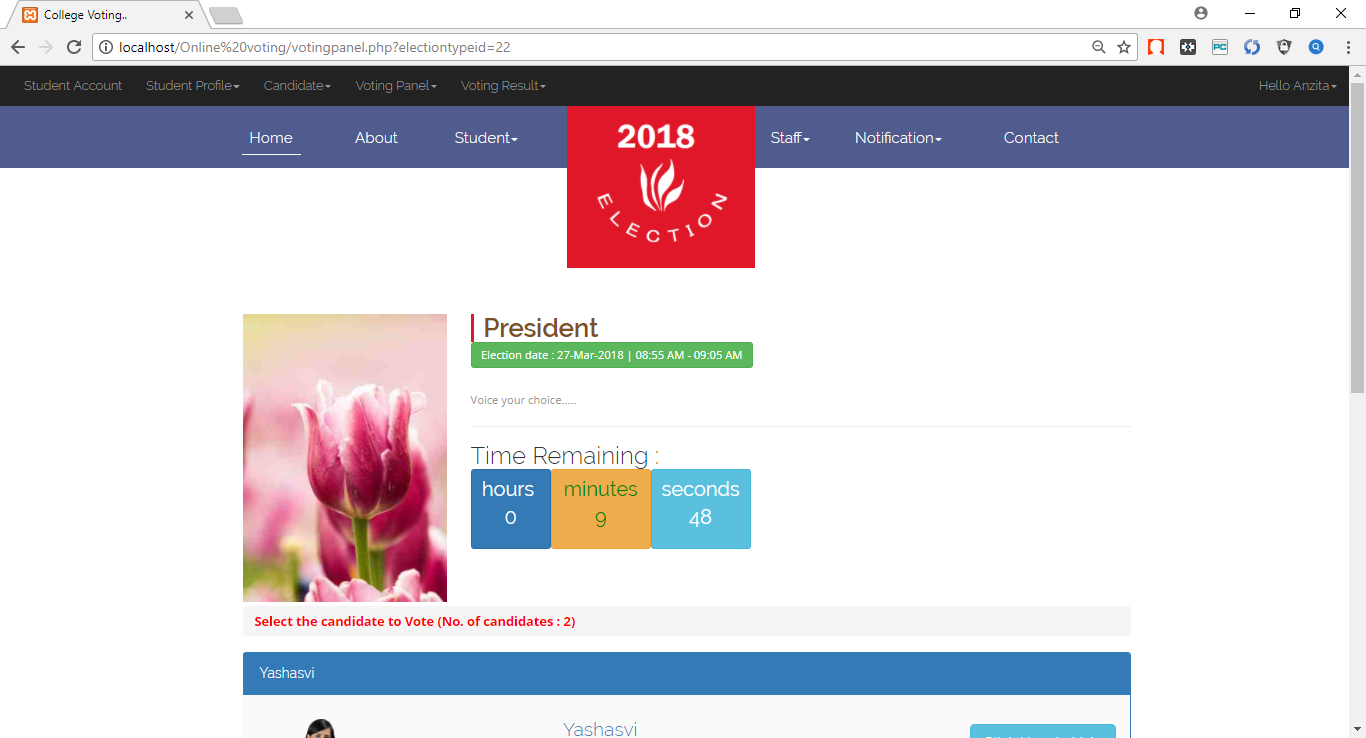


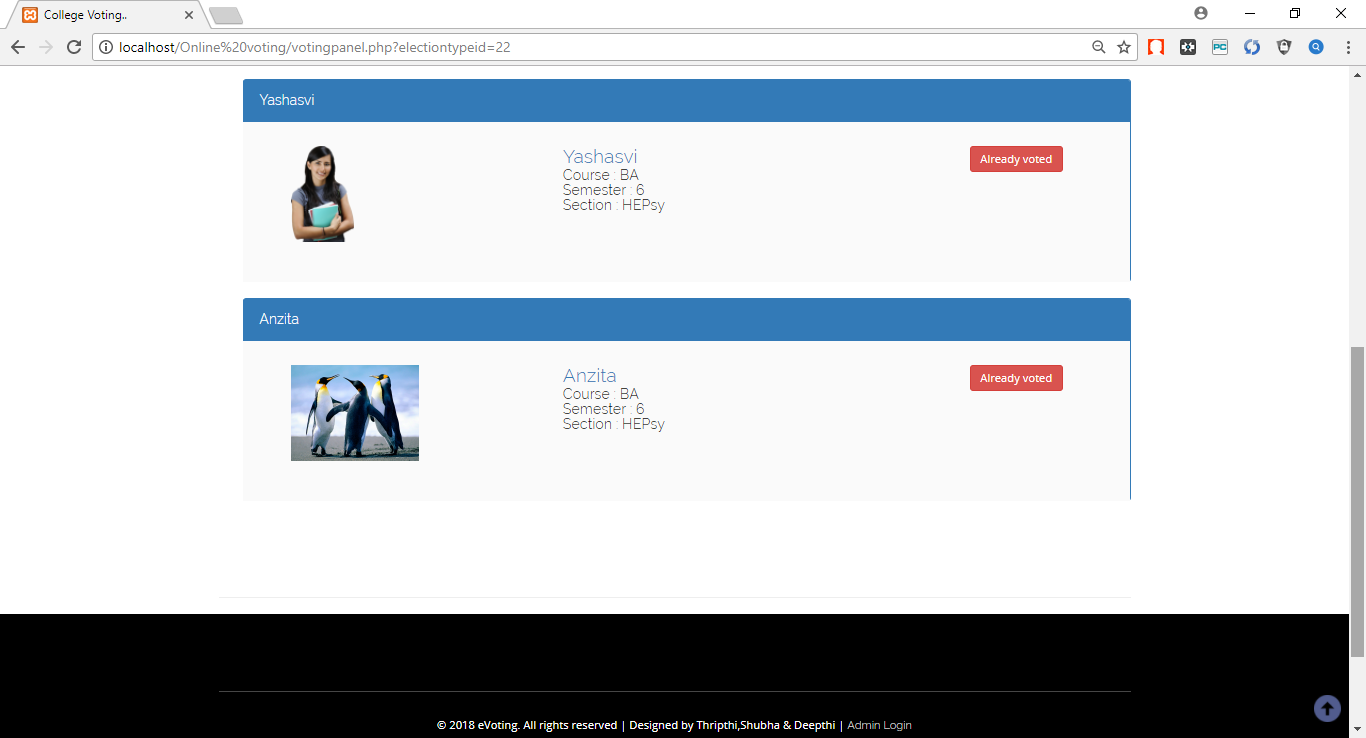




Here the student apply for election.

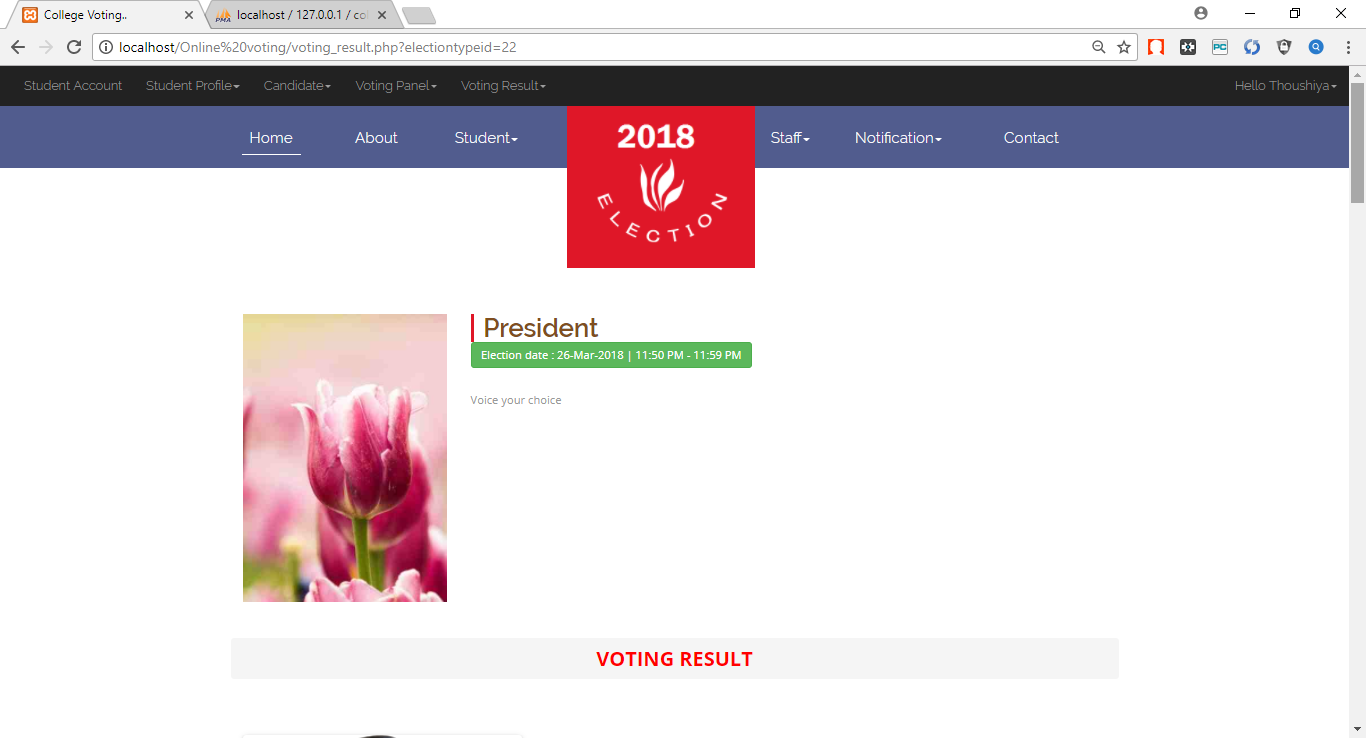
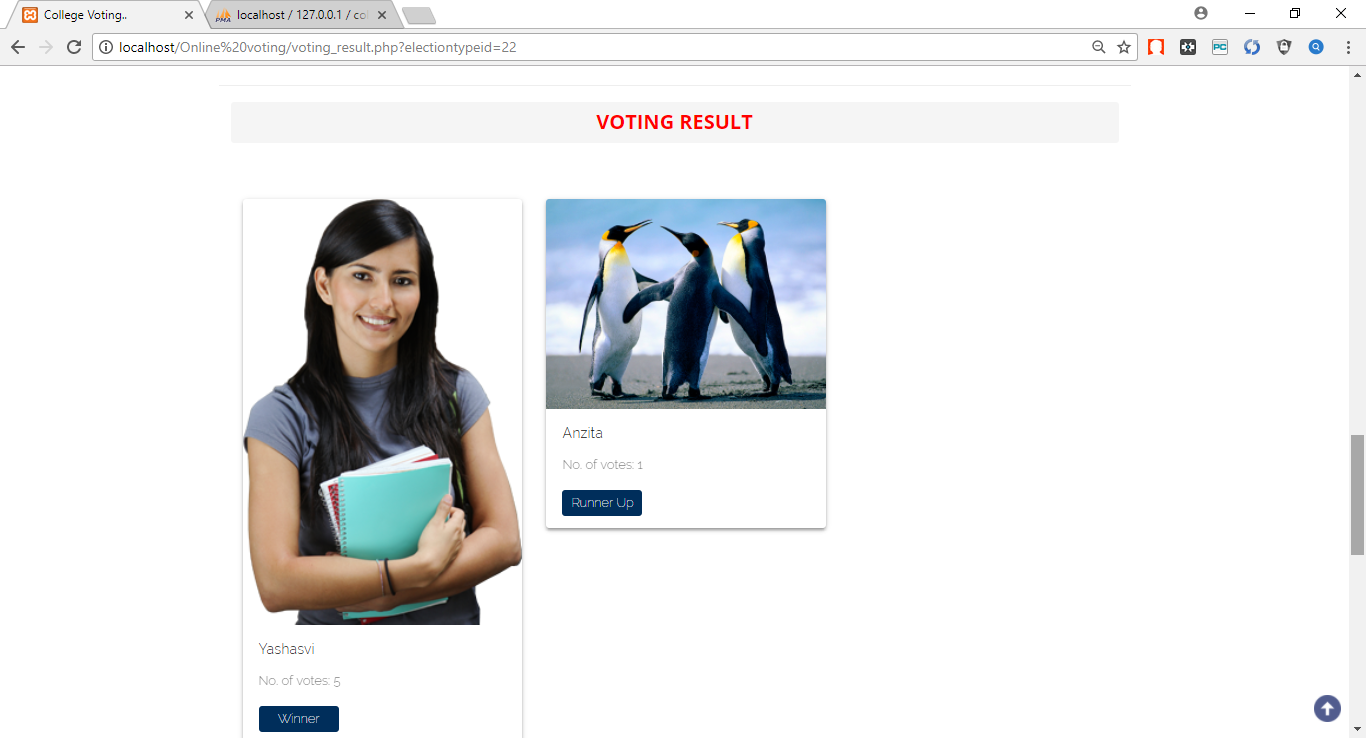
* **Voting panel**

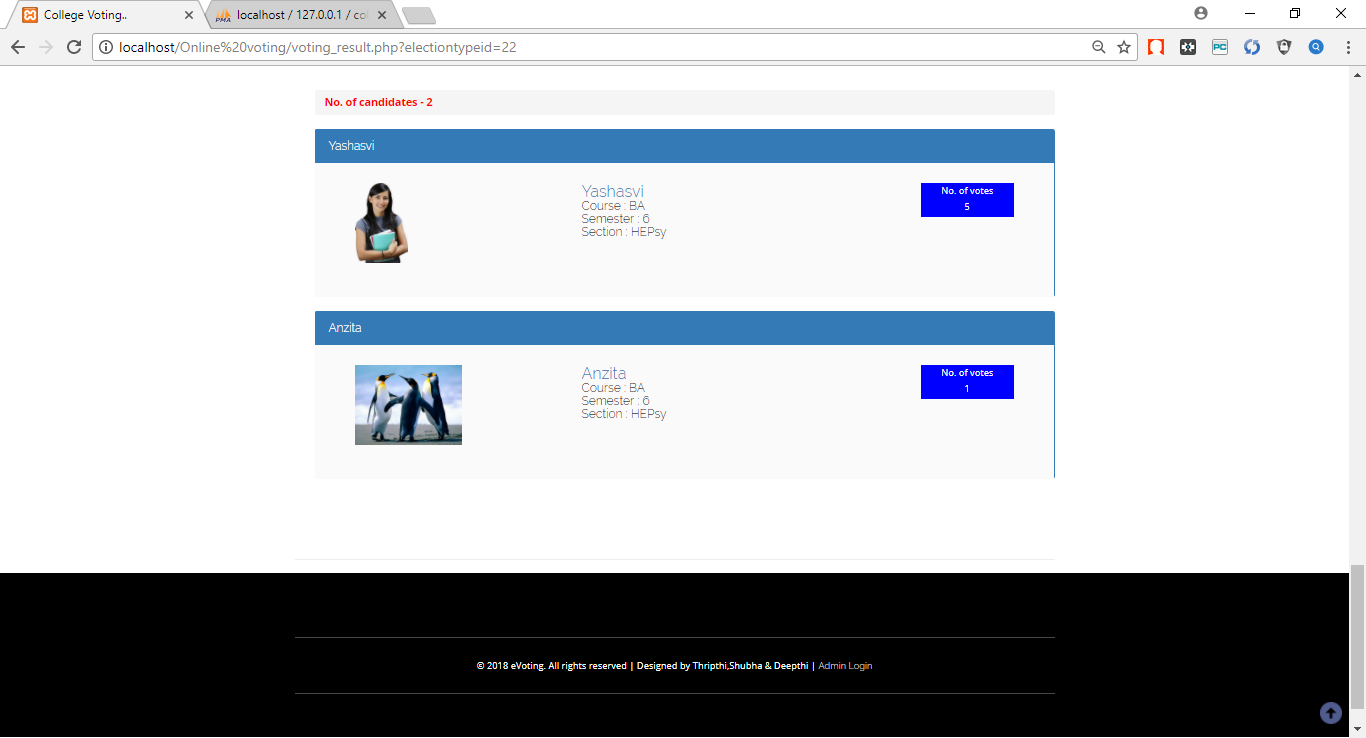




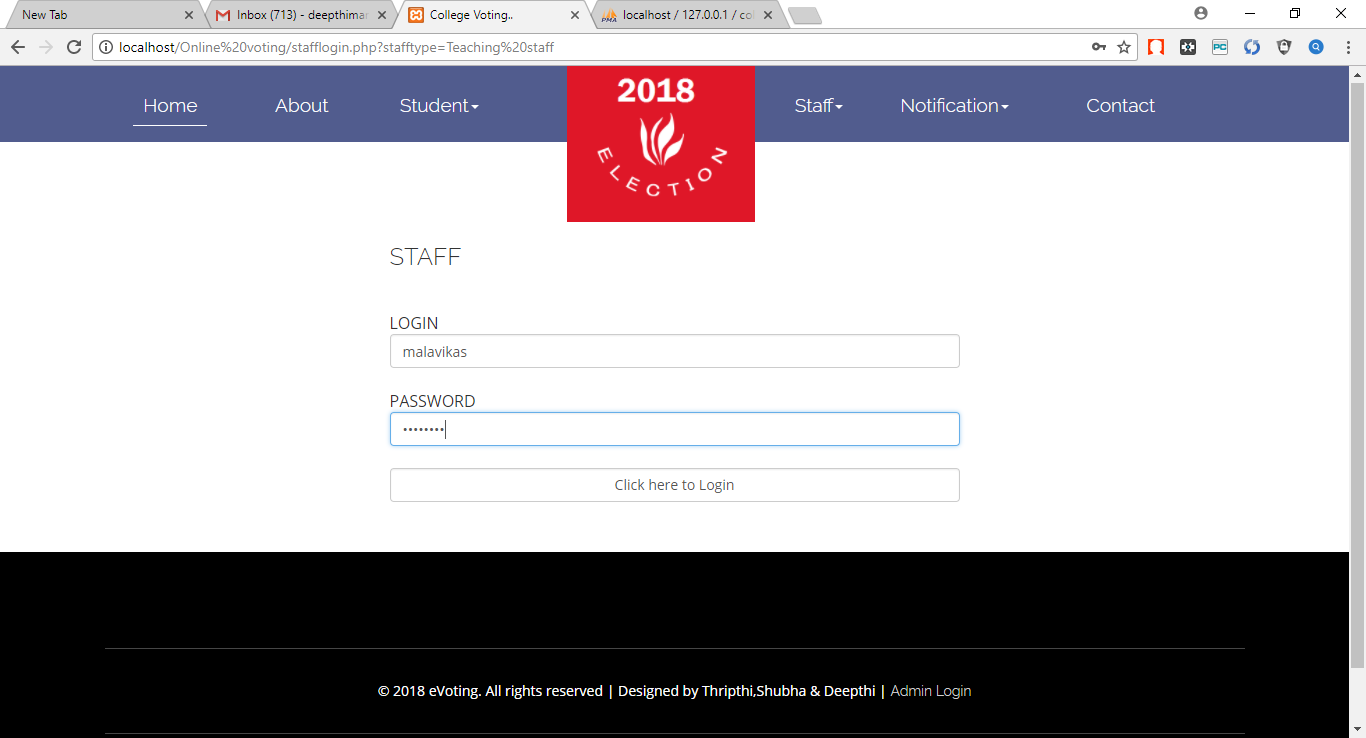
Here the student vote for the candidate.

* **Voting result**

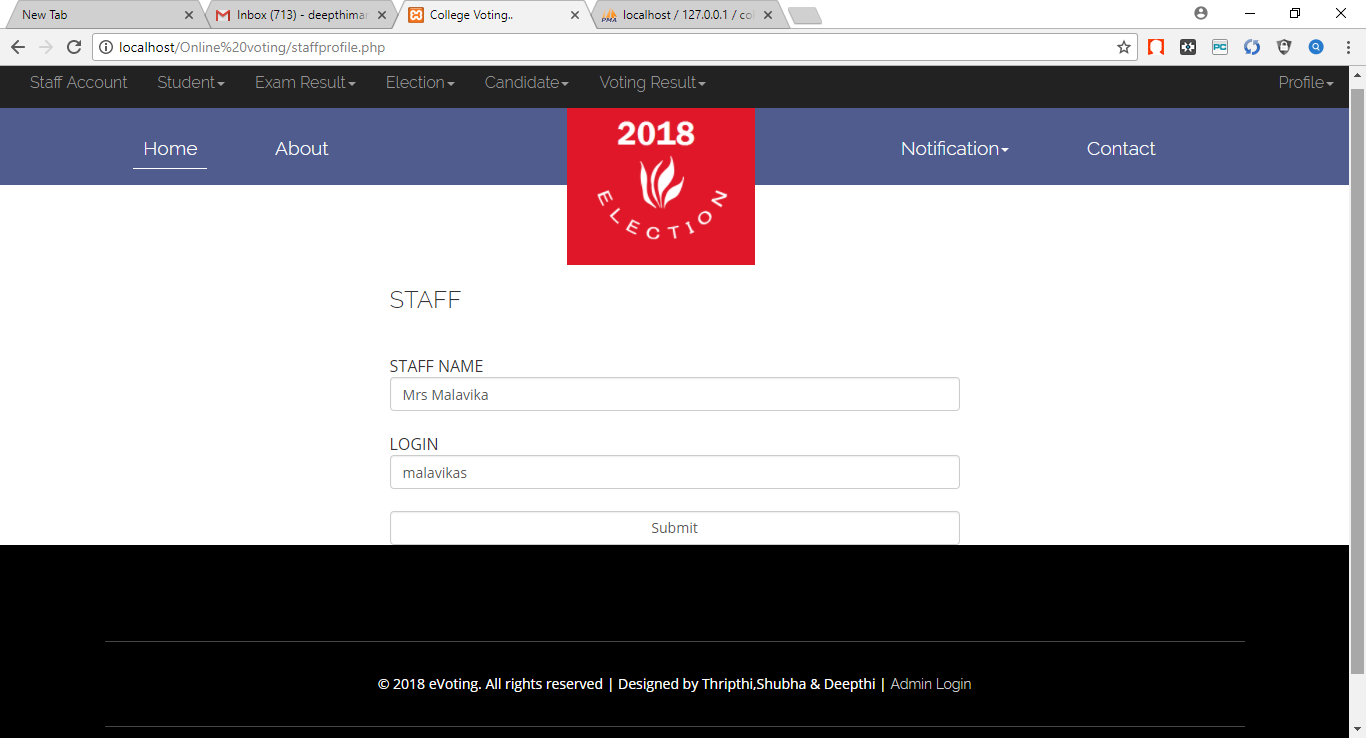




Result is published here.

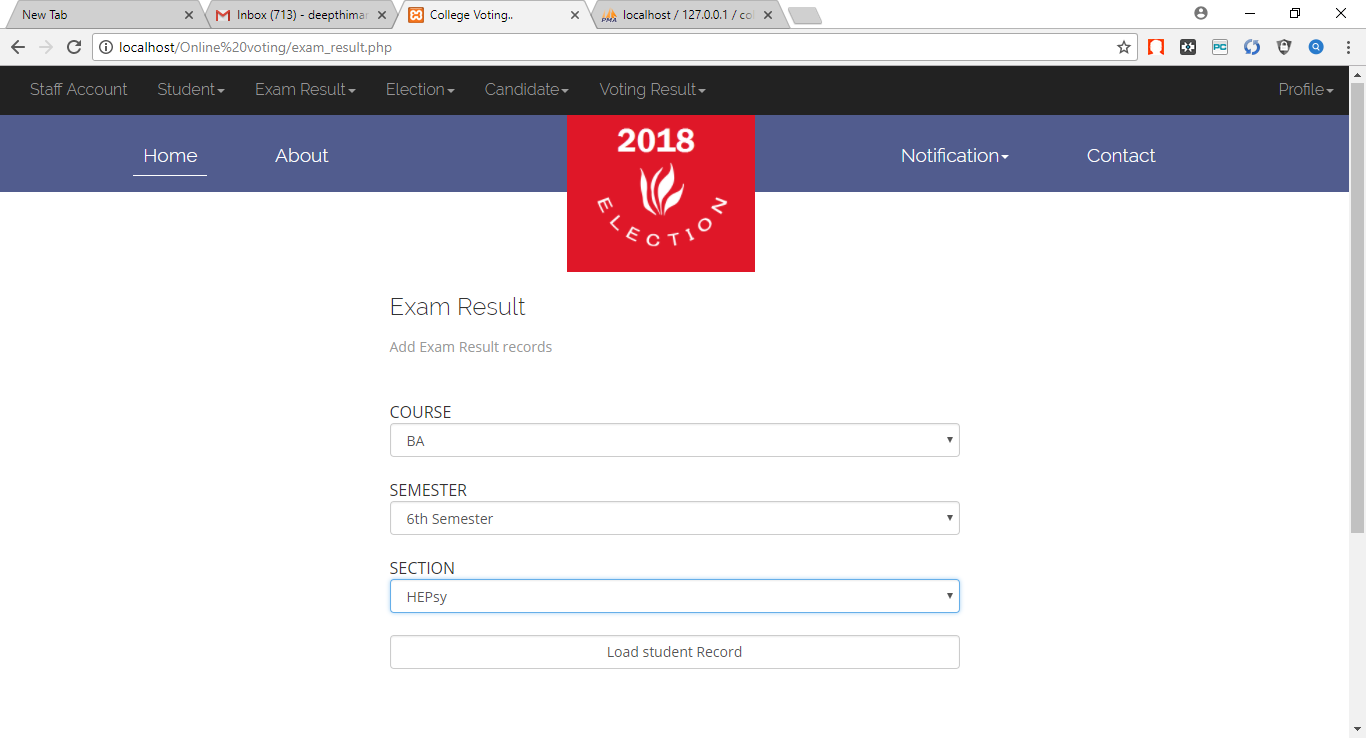
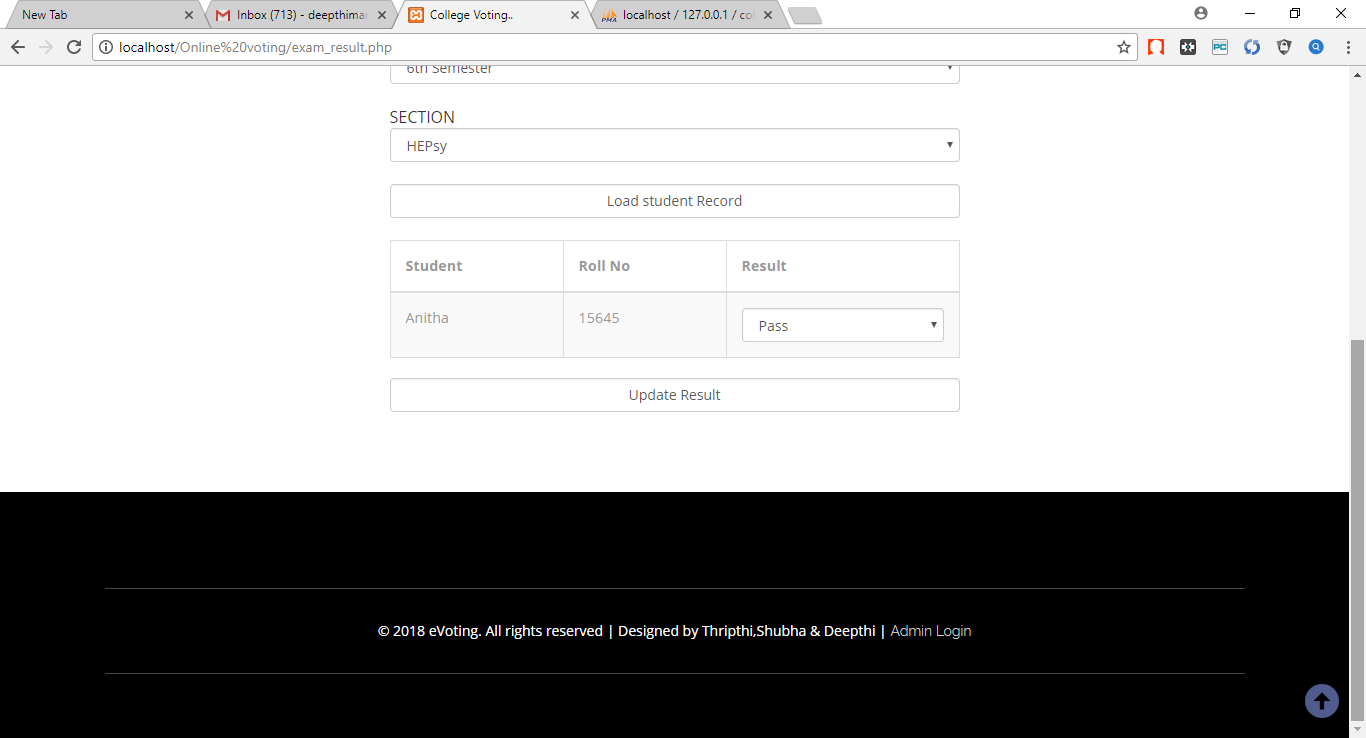
* **Teaching staff login**

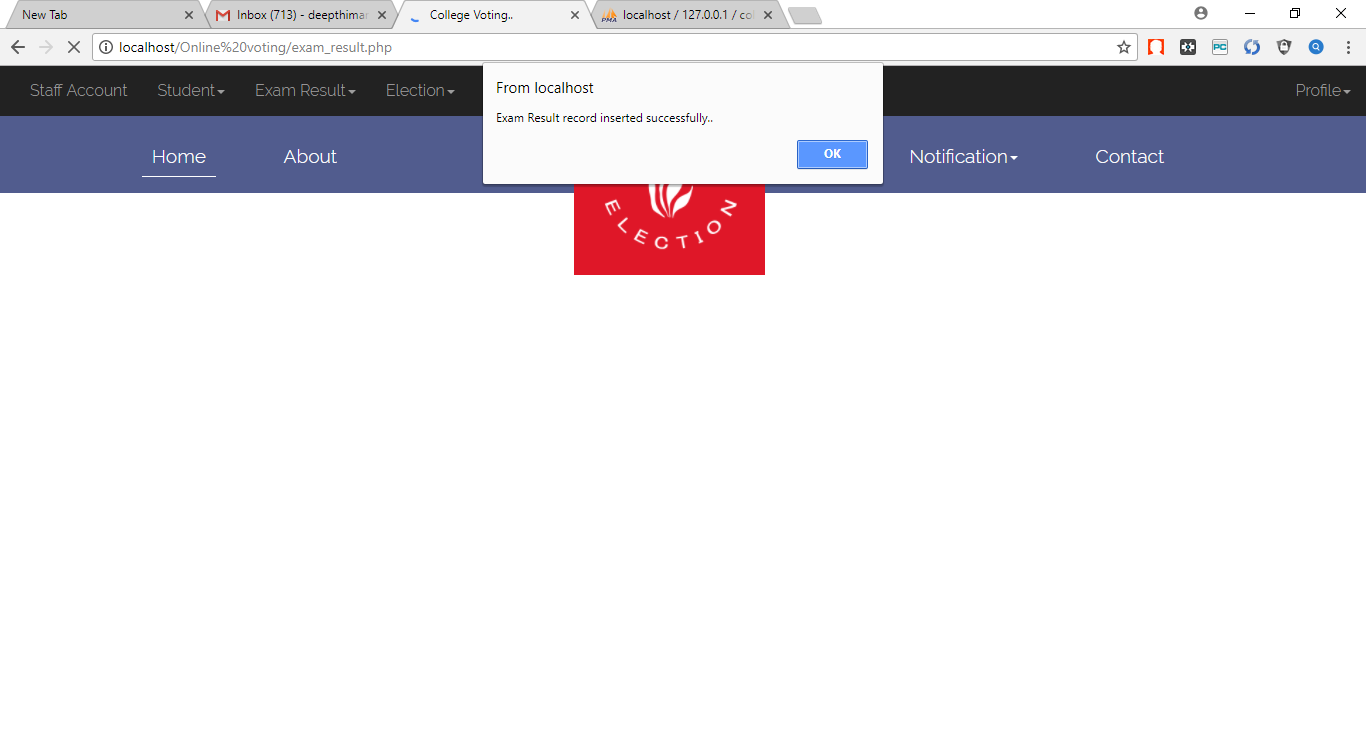
Teaching staff will use this login panel to use the site. Authorized teaching staff will login successfully.

* **Teaching staff profile**

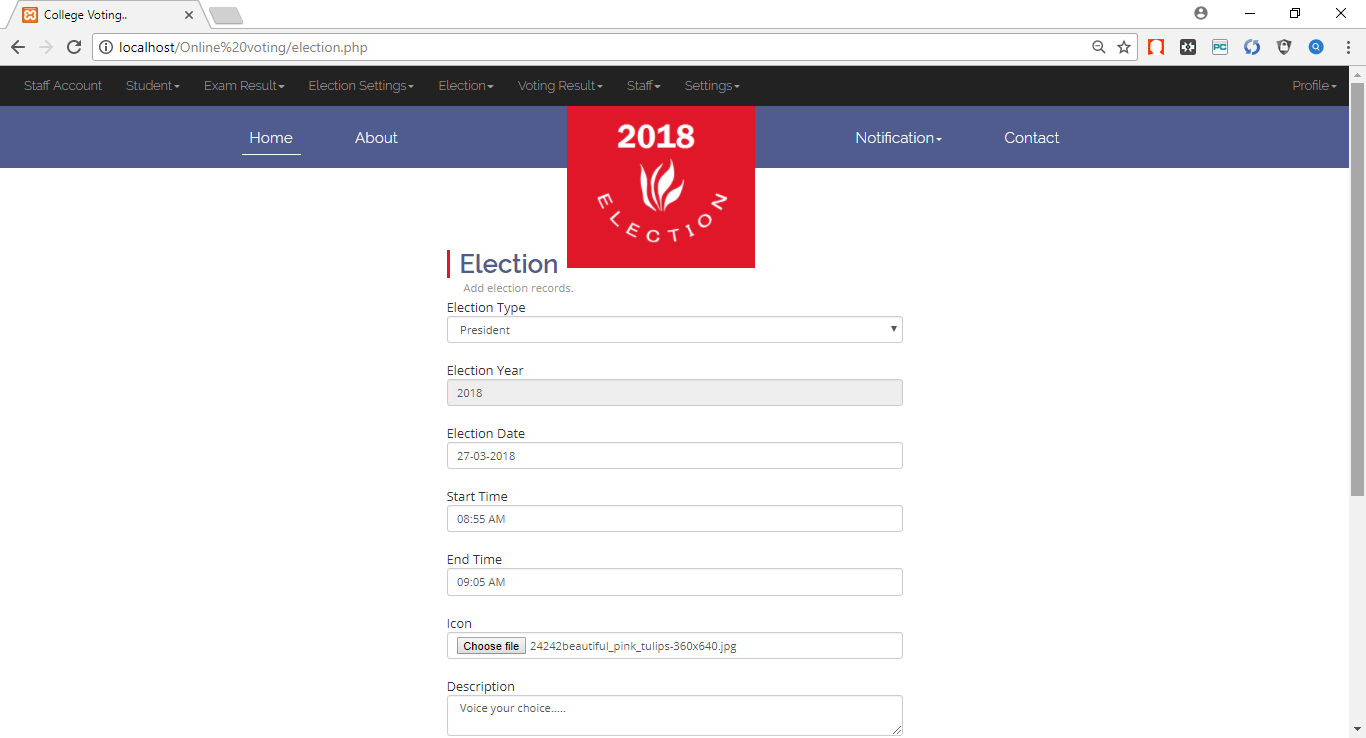
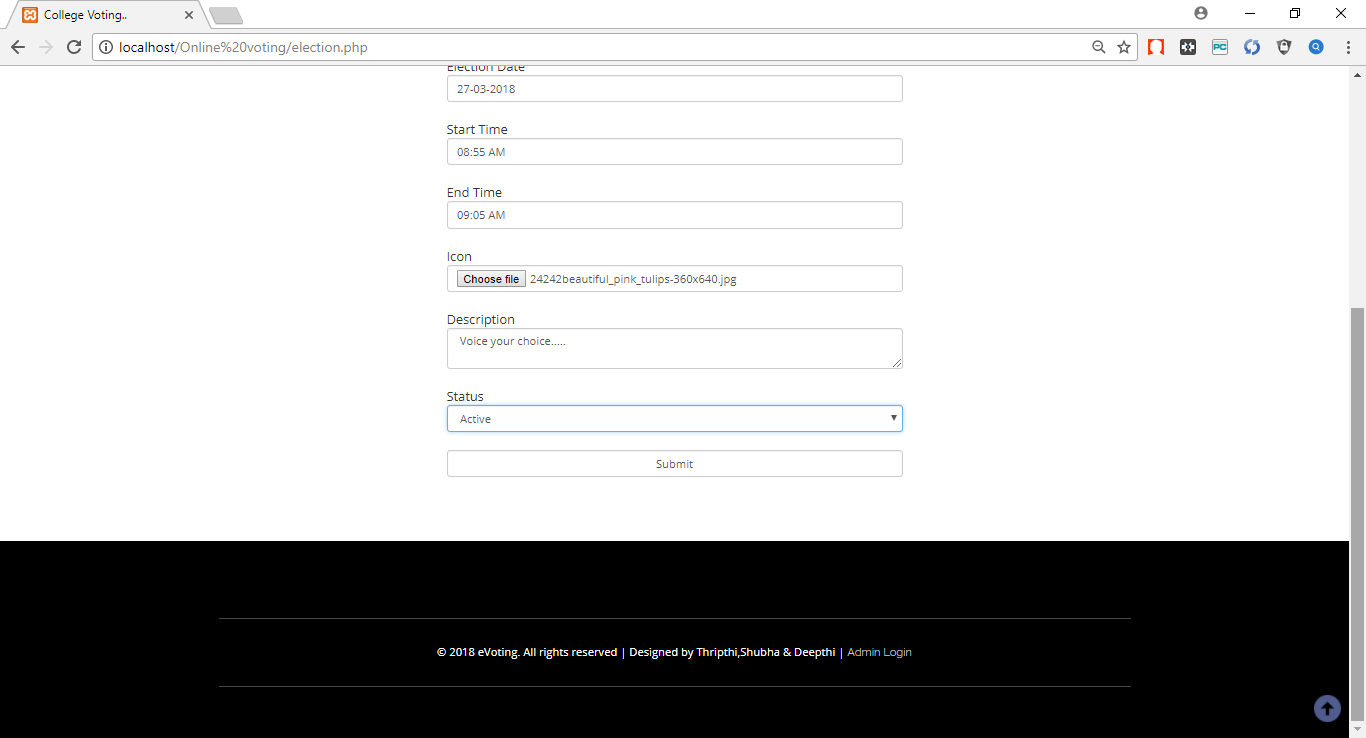
Profile which shows the teaching staff details.

* **Adding exam result:**

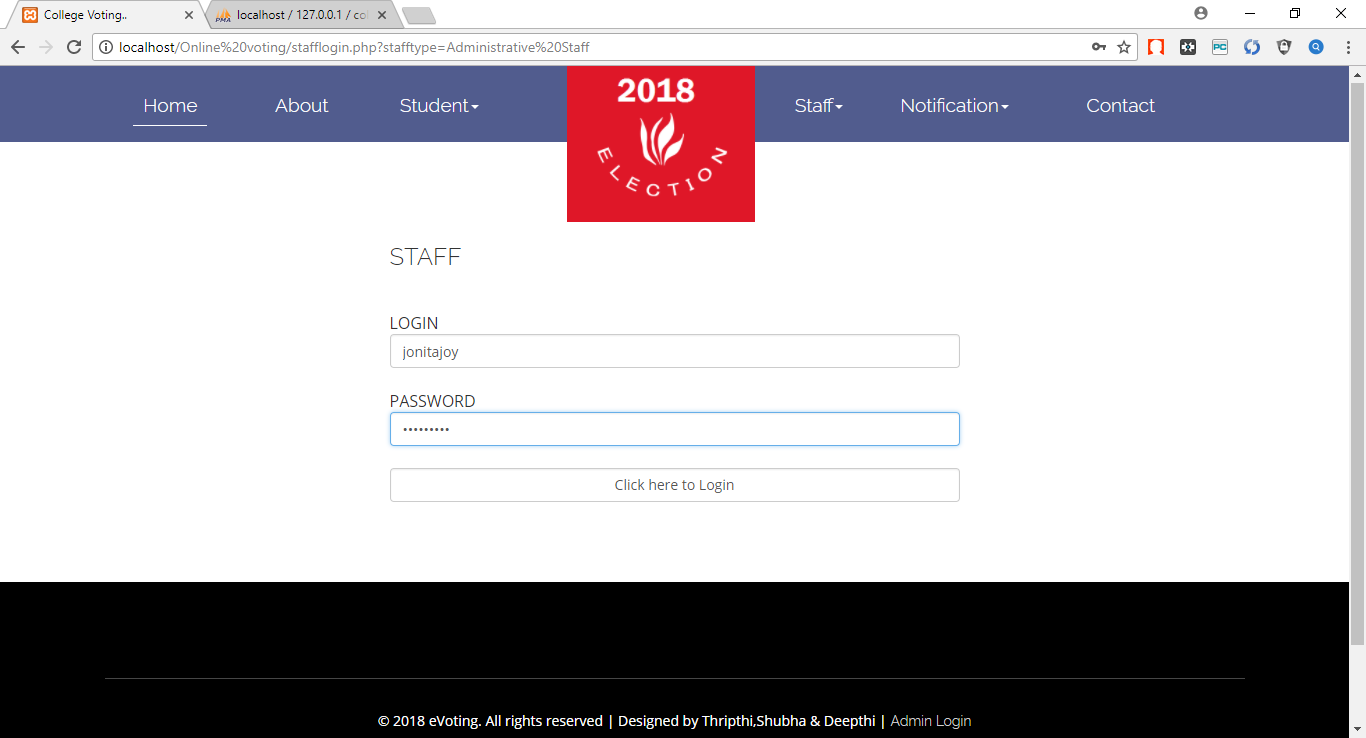




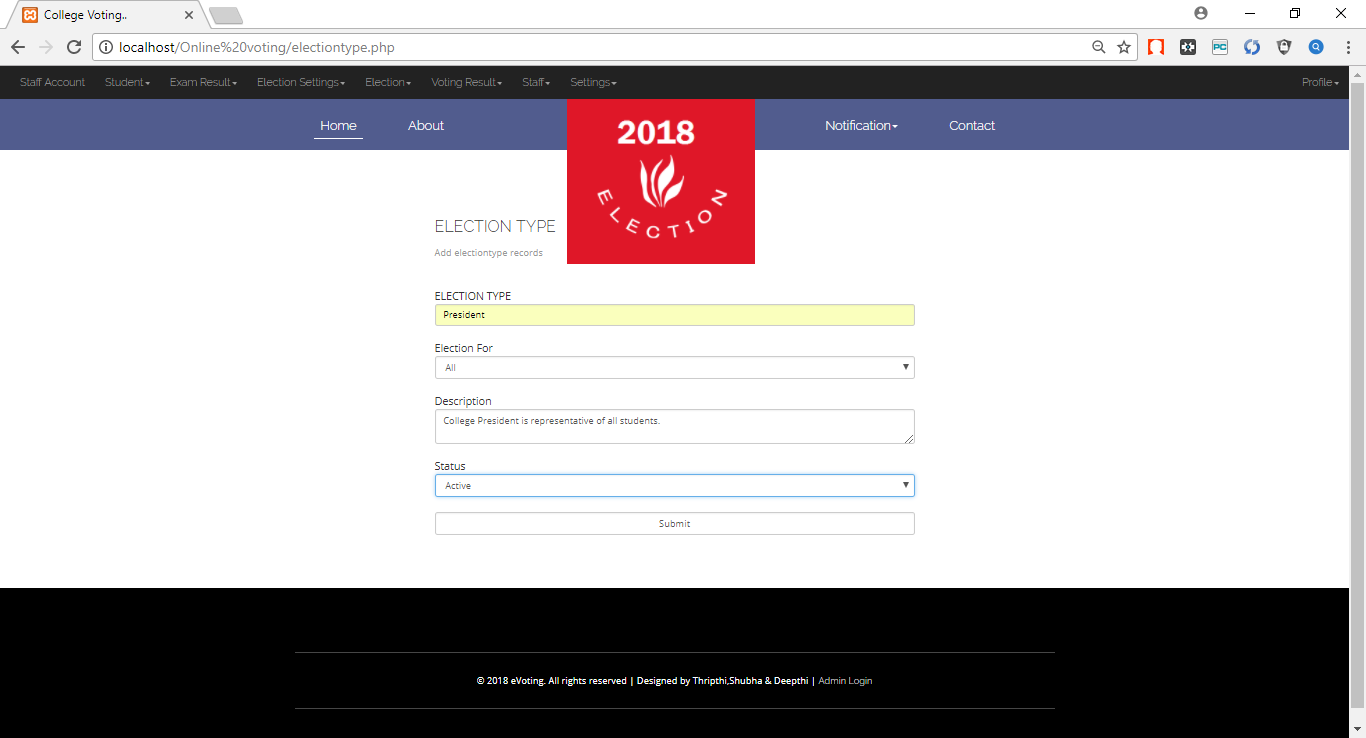
Here,the teaching staff uploads the student result. The student can contest for the election only if the result is pass.

* **Adding new election**

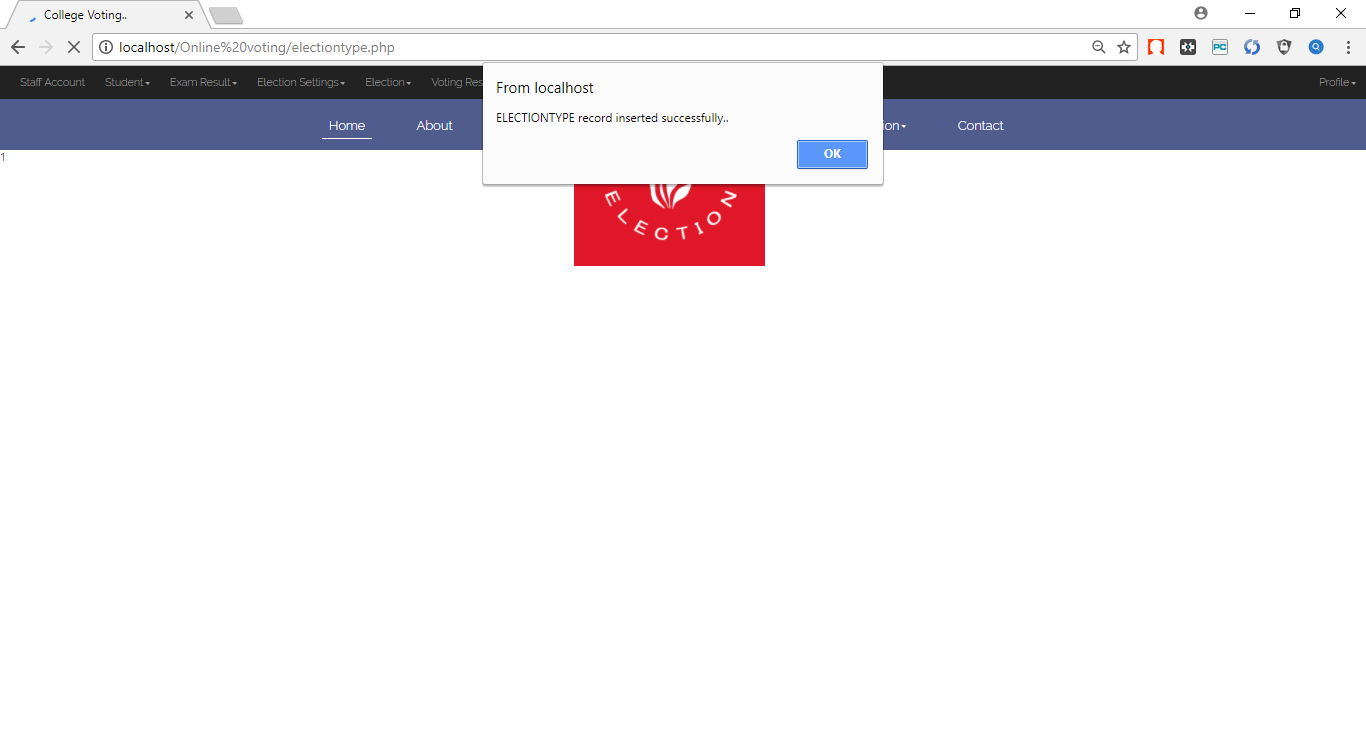
Adding new election.

* **Administrative staff login**

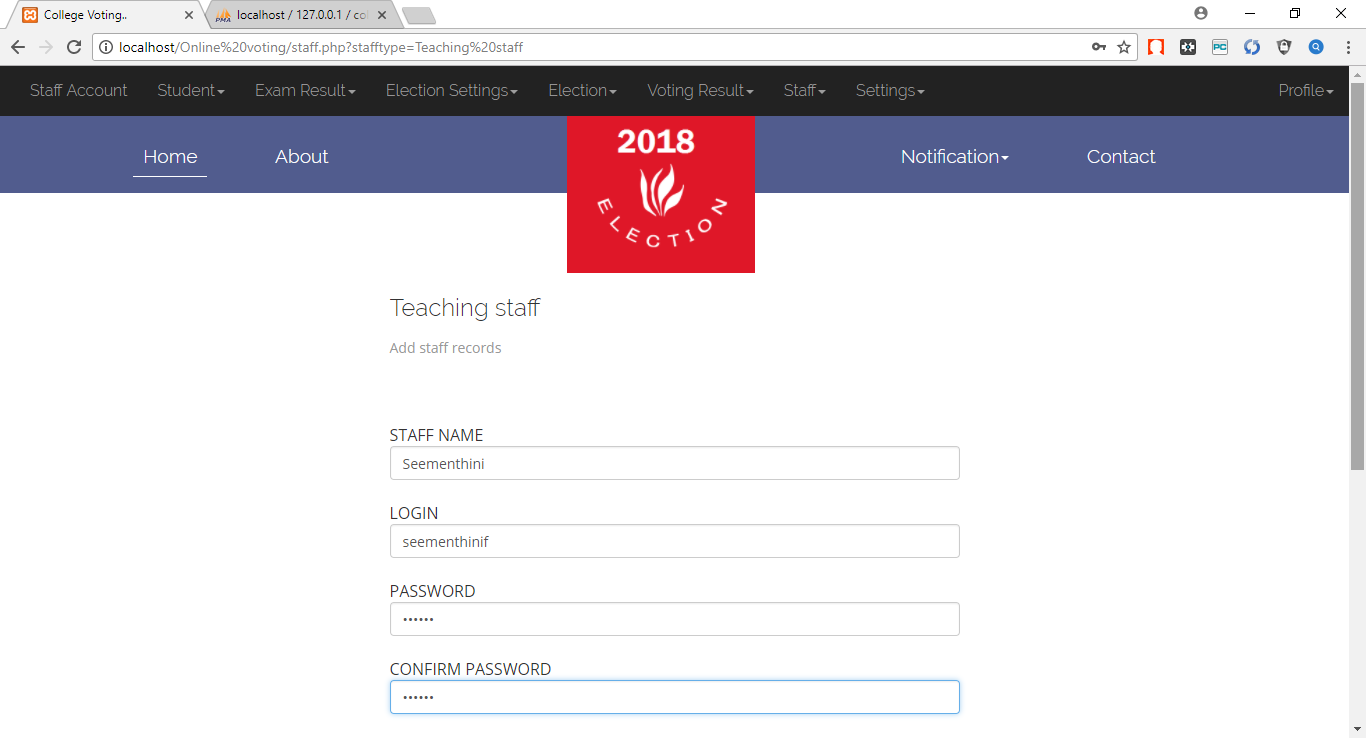
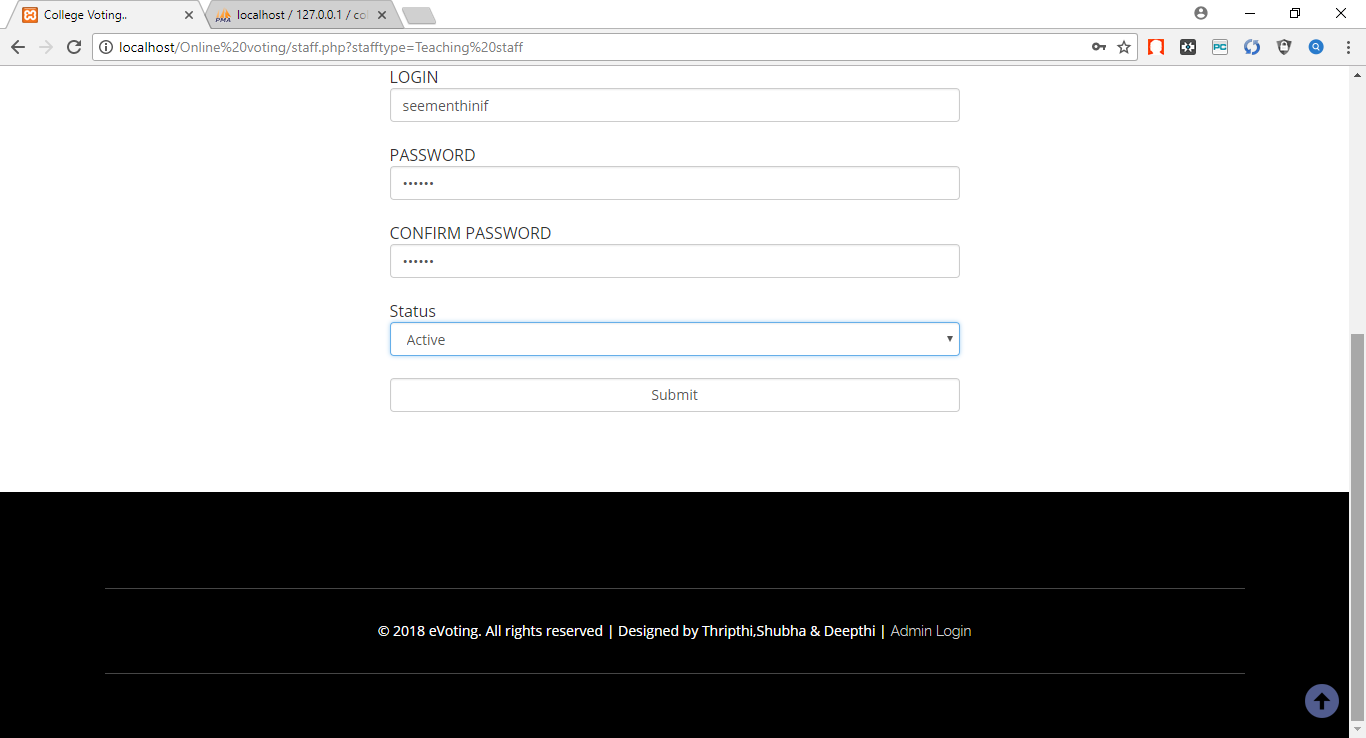
Here the administrative staff is logging in.

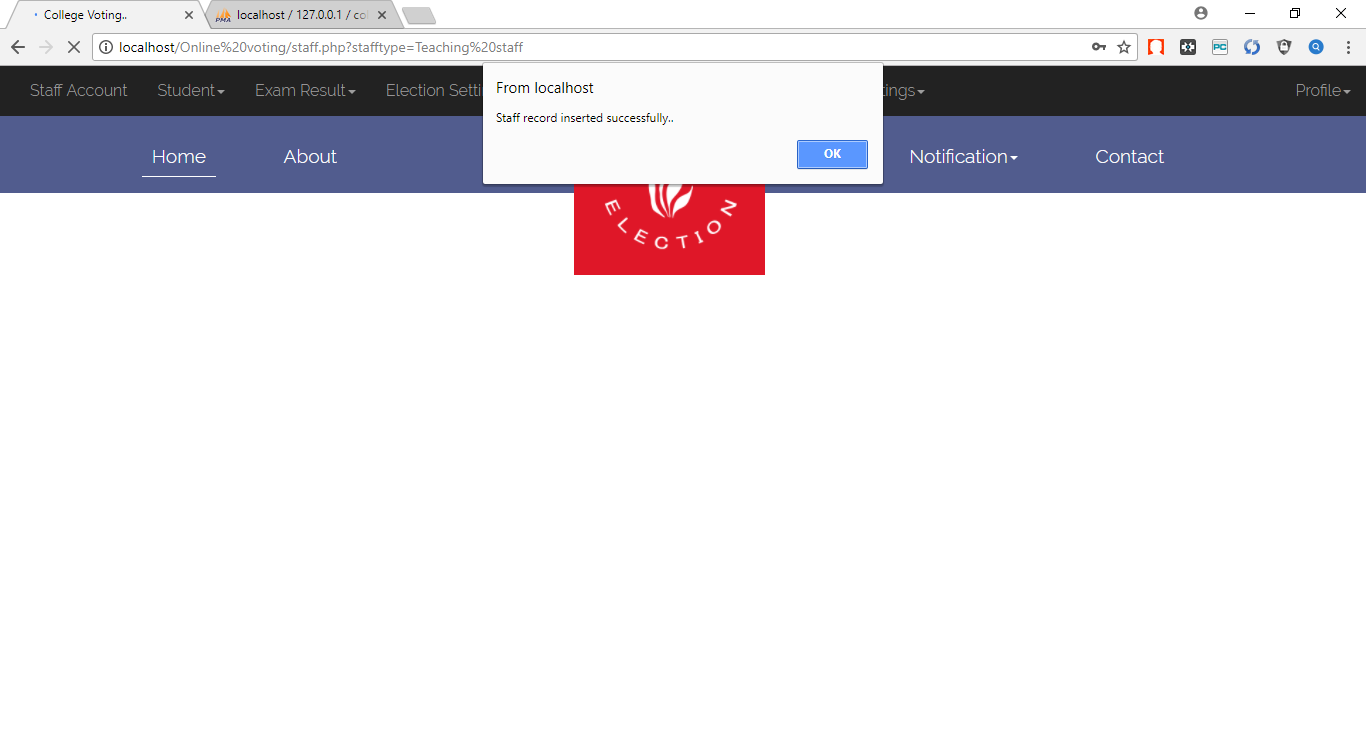
* **Adding election type**

Here we are adding election type.

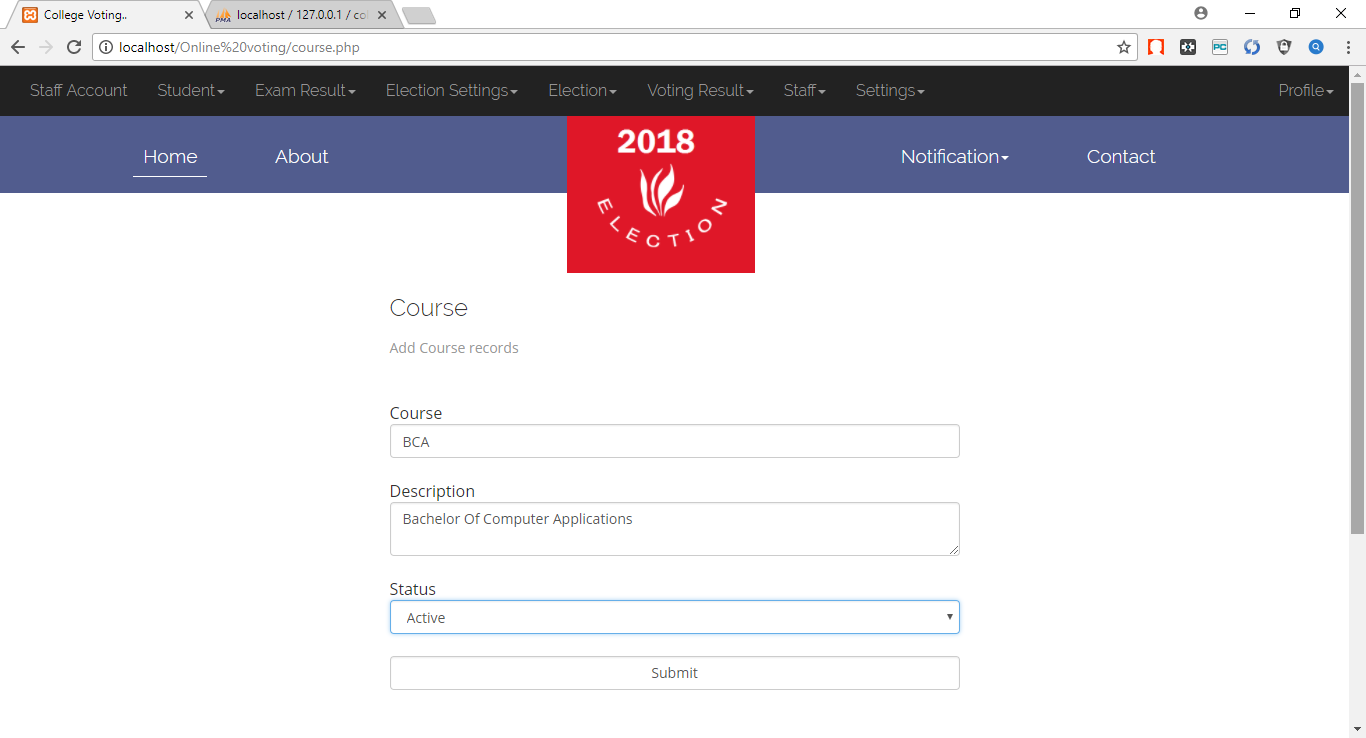


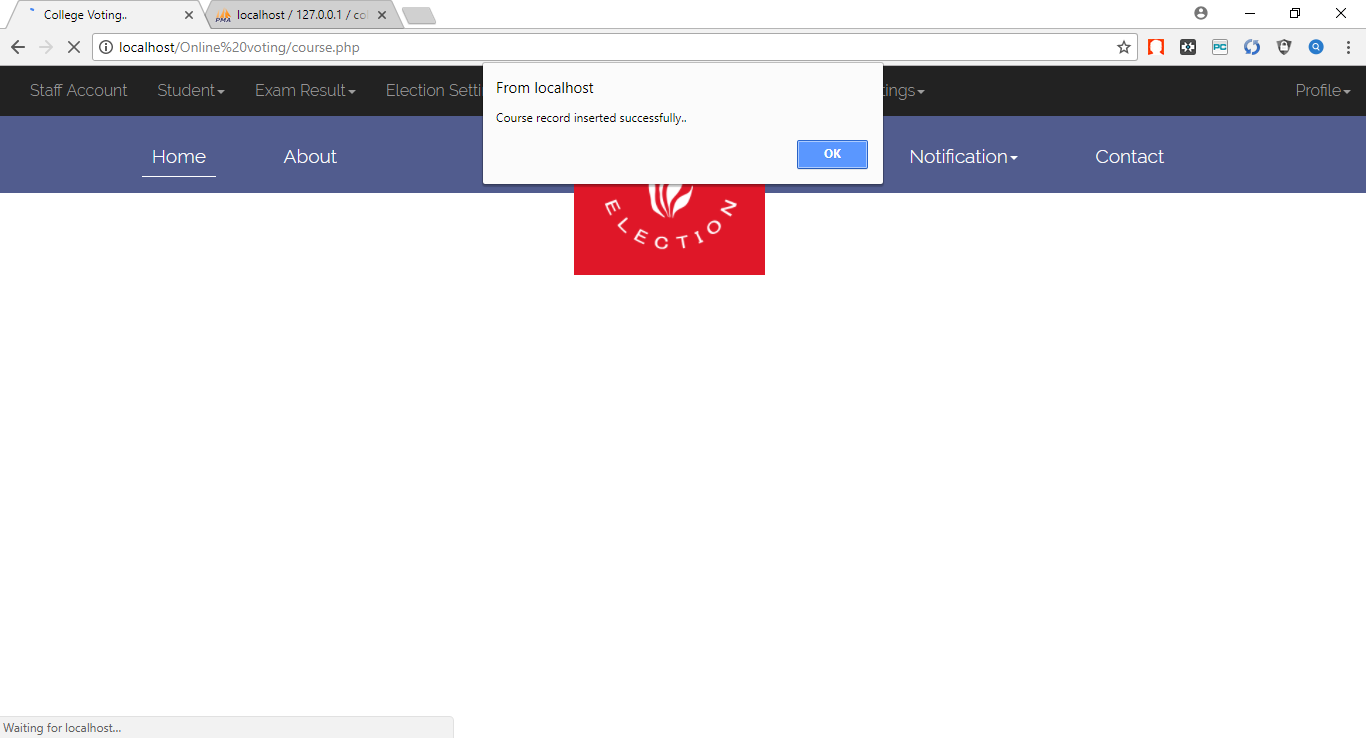
* **Adding teaching staff**



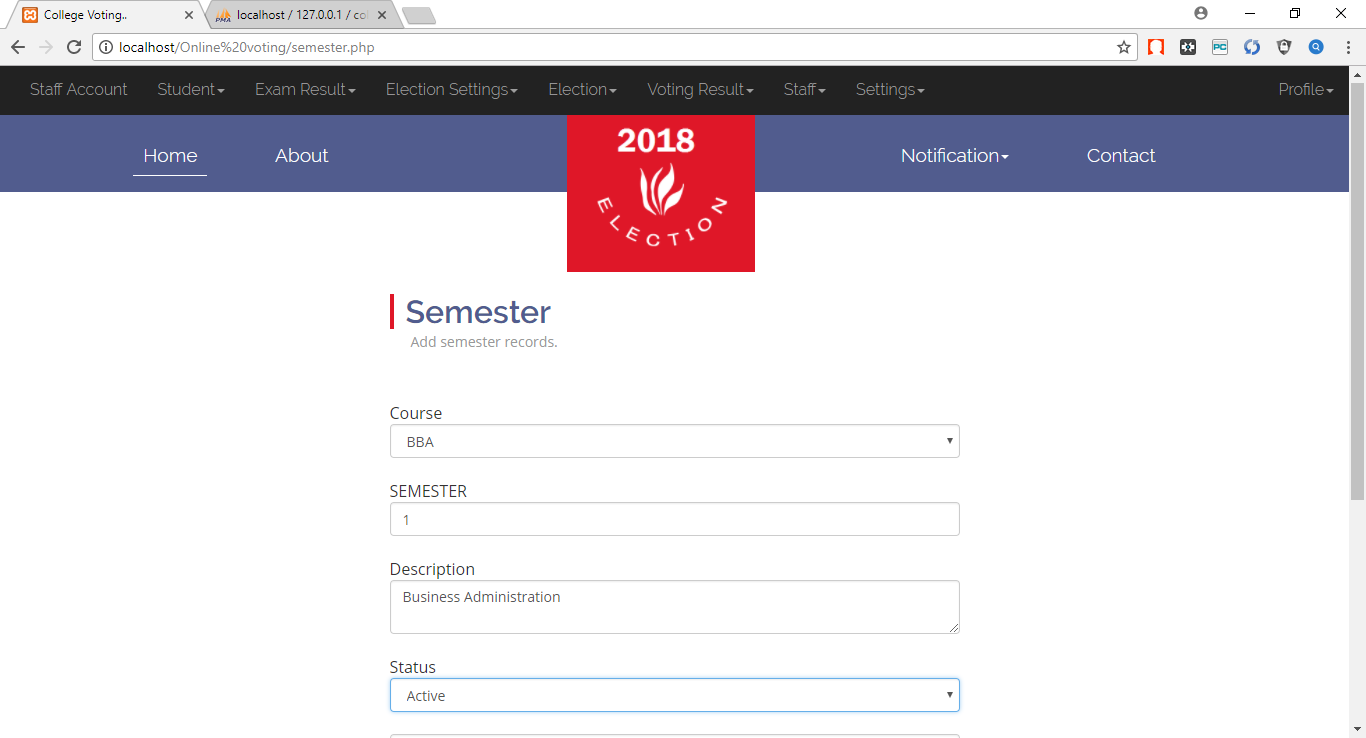
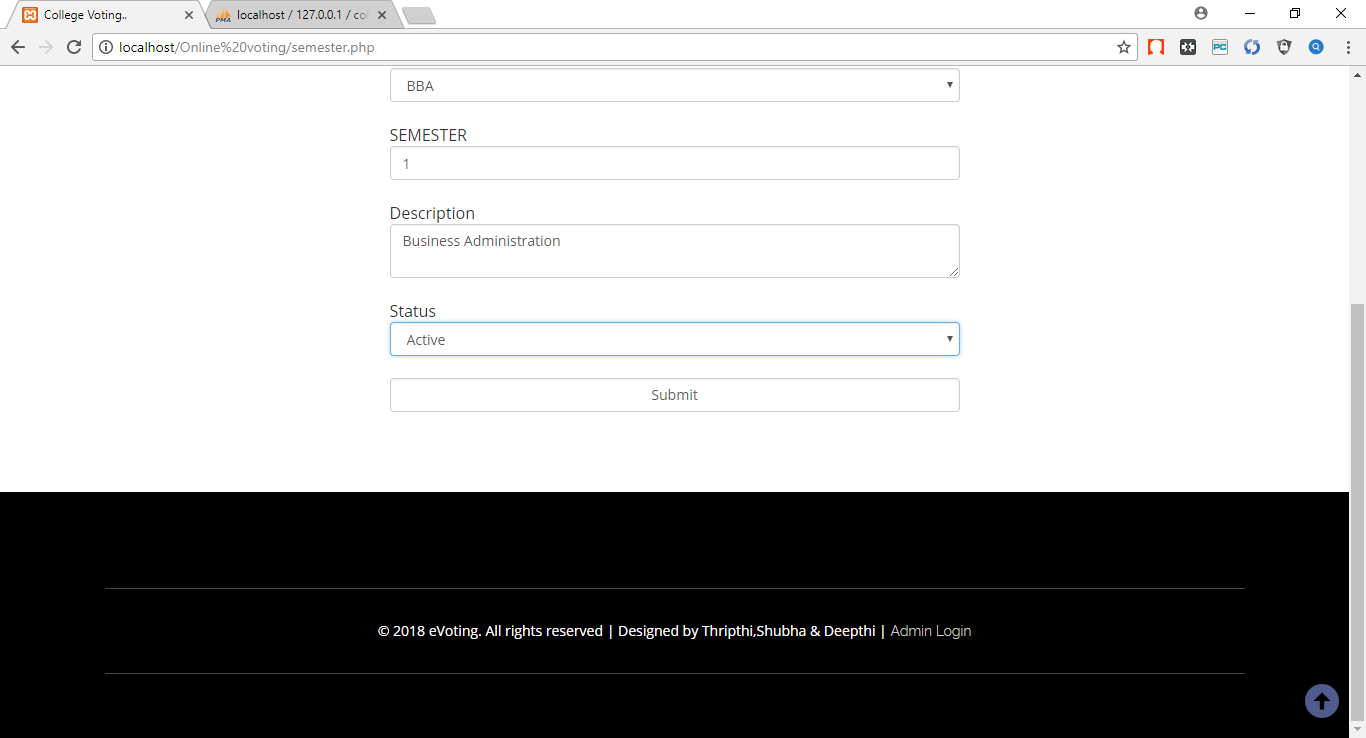


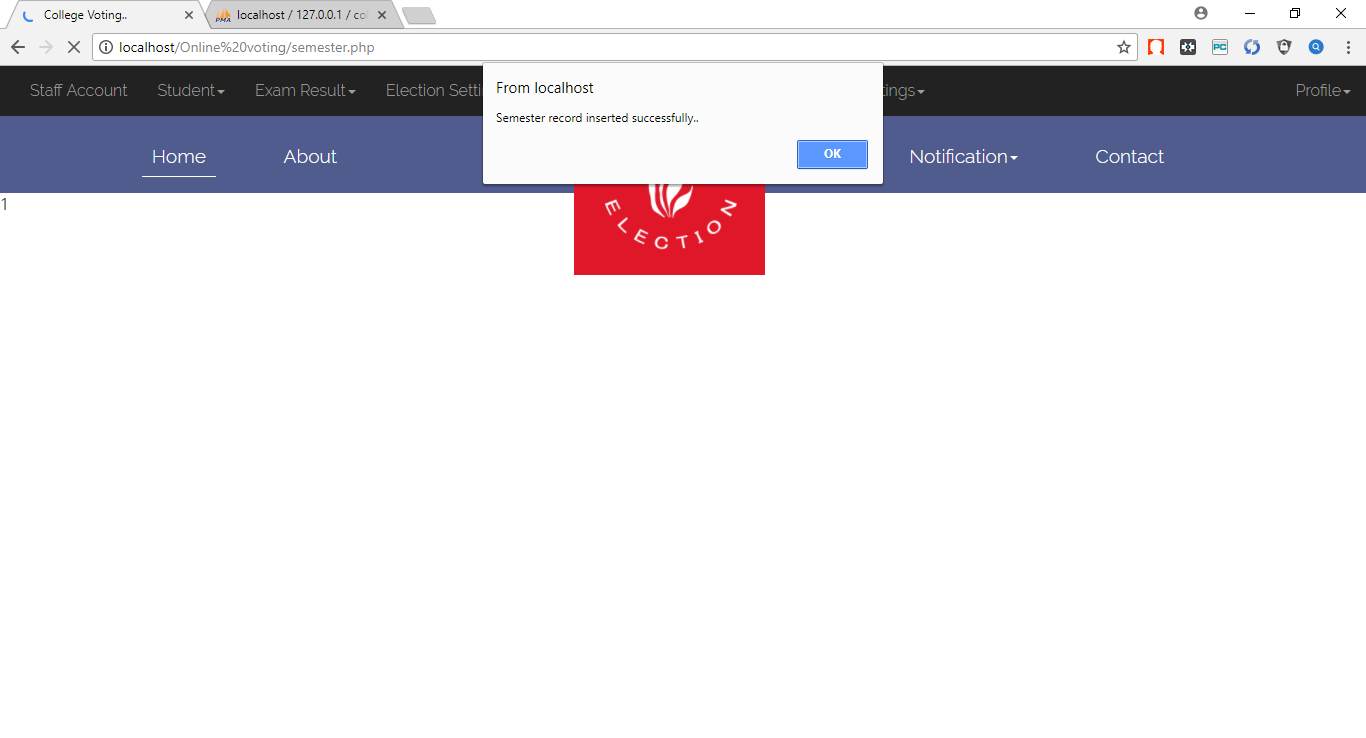
Here teaching staff records are inserted.

* **Adding course**

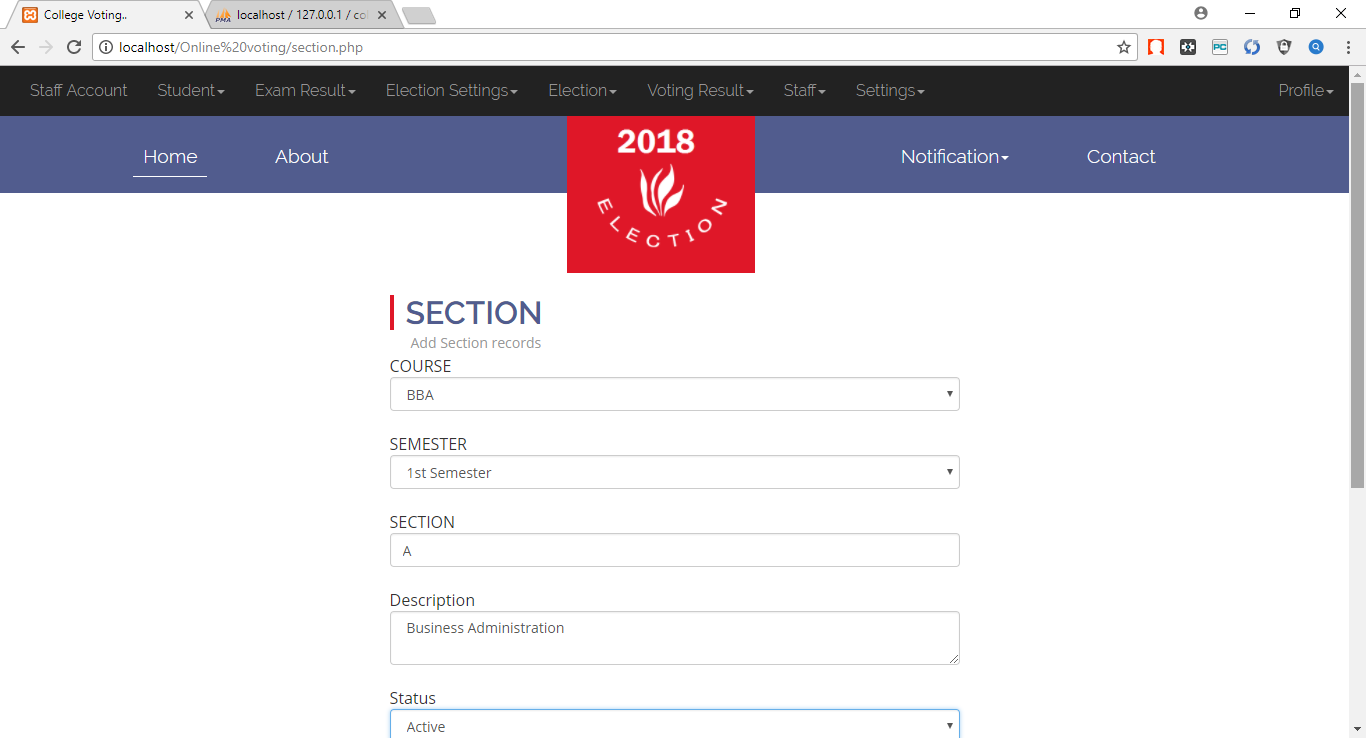
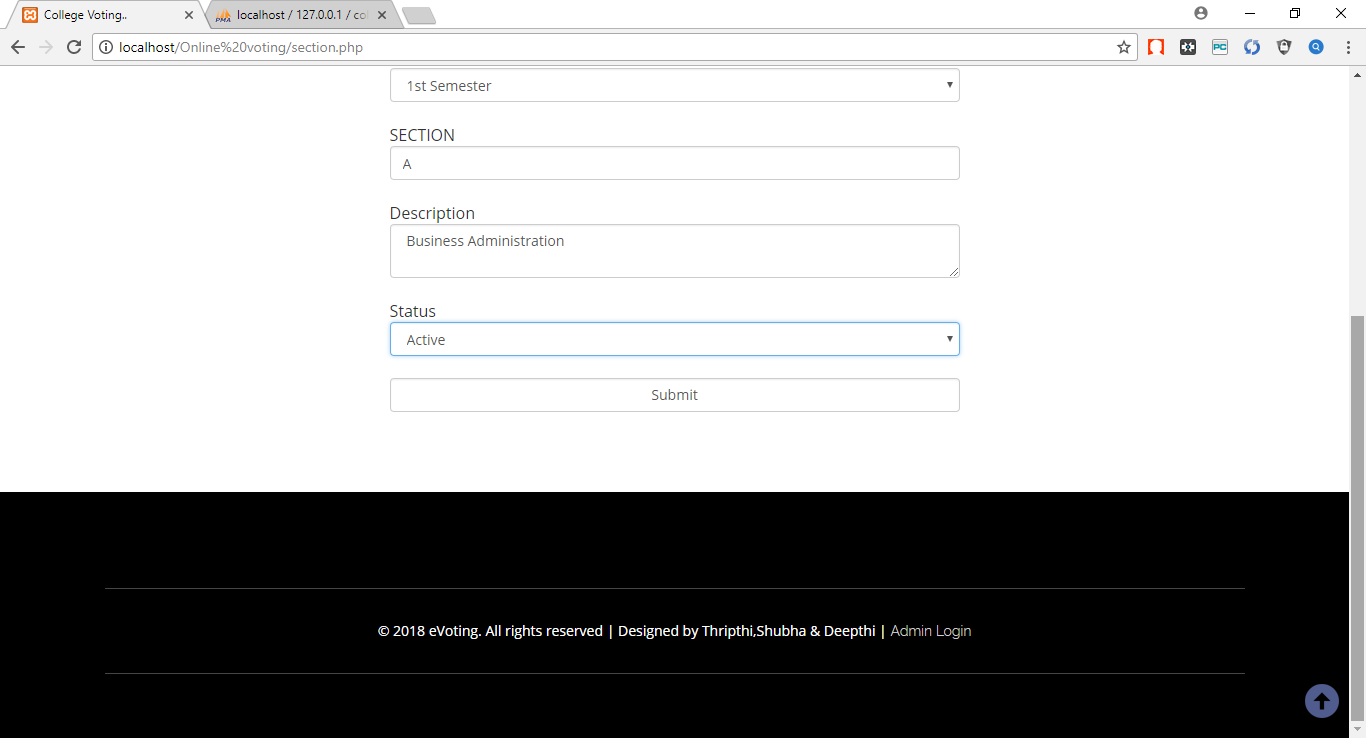


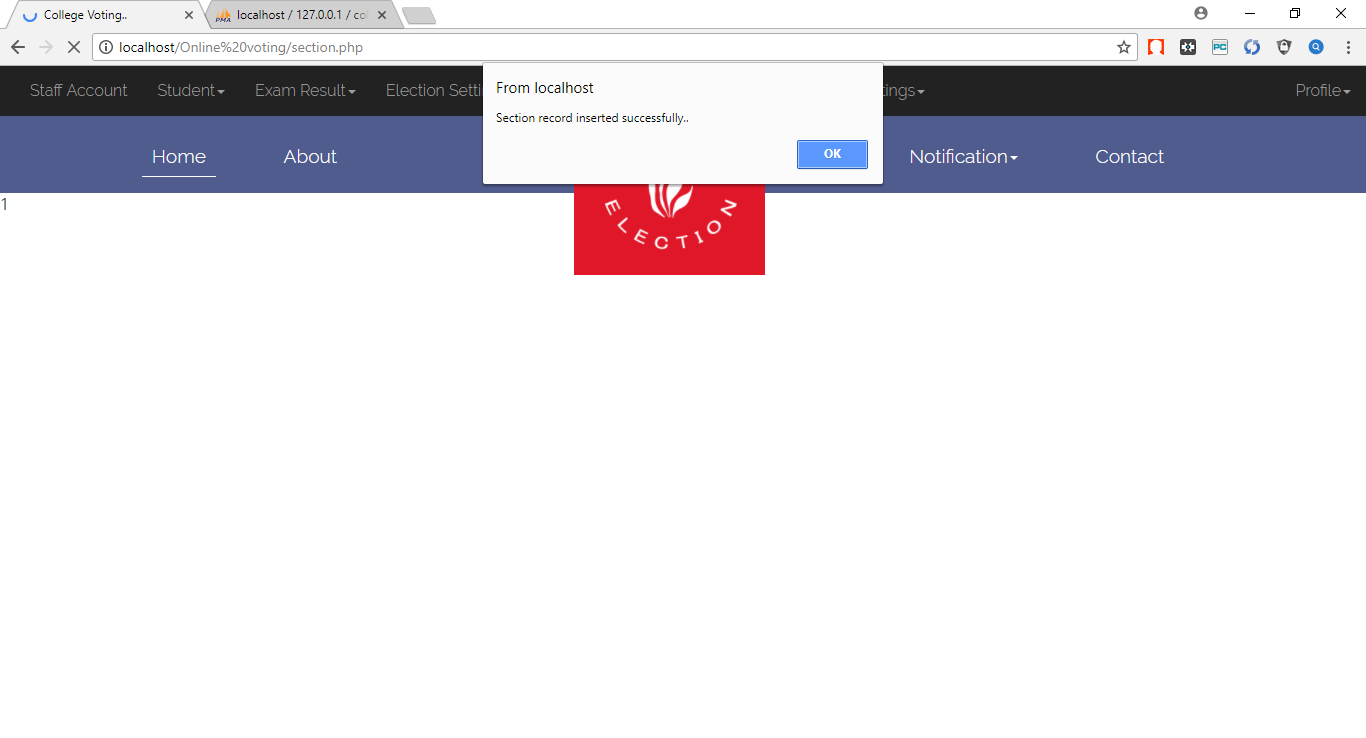
Here courses are added.

* **Adding semester**

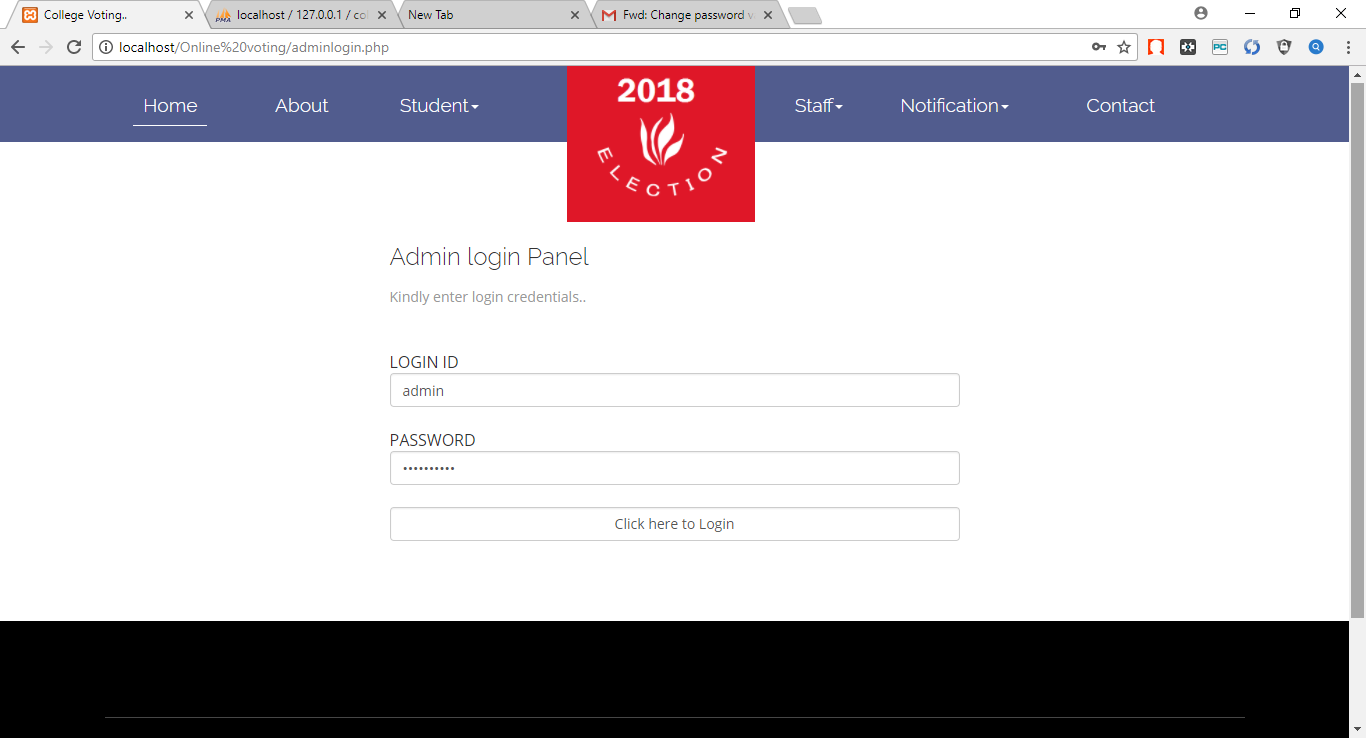


Here semester record is added.

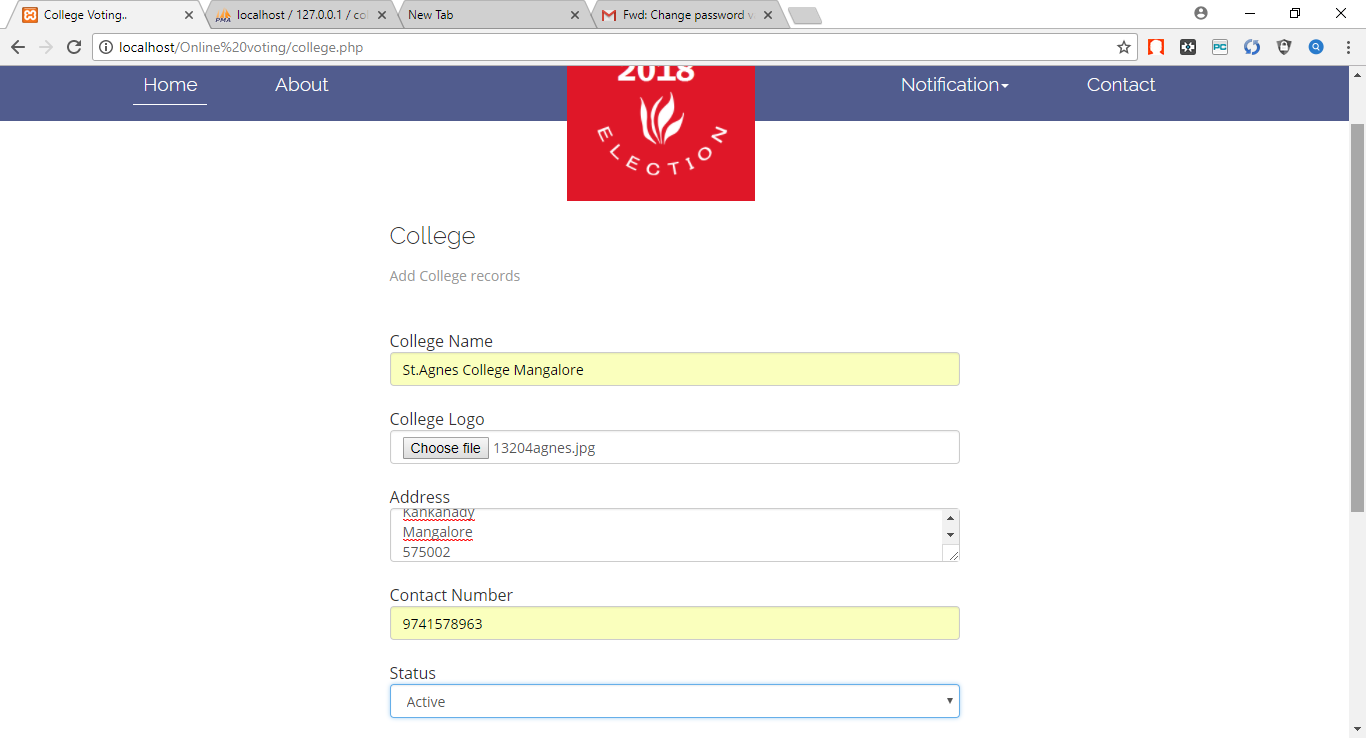
* **Adding section**

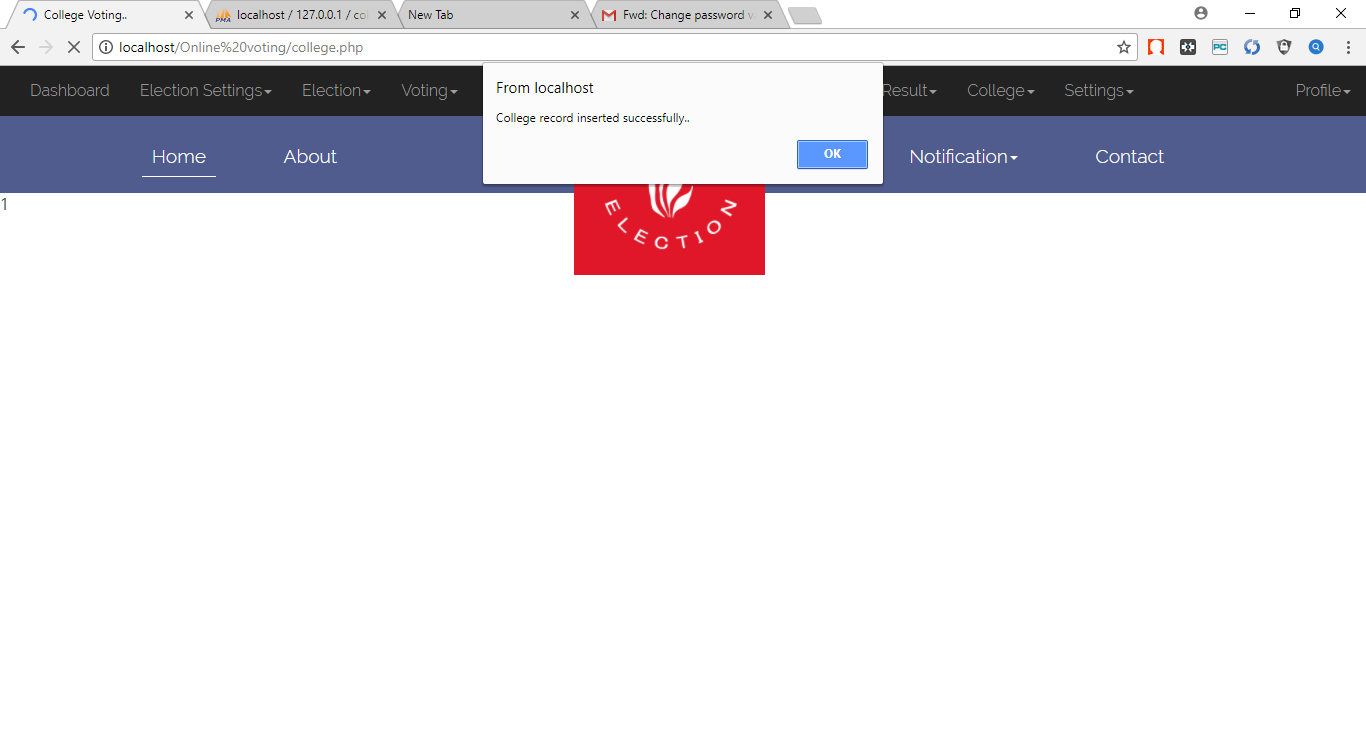


Here section record is added.

* **Admin login panel**

Here Admin logs in.

* **Adding College**



College records are added.

Conclusion

Conclusion

The project work titled “College Voting” has been designed using PHP Hypertext Pre-processor wherein many user friendly form controls have been added in order to make it a user interactive application. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The system forms a general platform for building most advanced insurance system. The future enhancements to the system can be made as technology changes.

**Future Enhancements:**

* In future we can build a mobile application for smart phones.
* Currently the system supports only desktop pc and in future we can make it mobile browser compatible site.
* The project can be made more user friendly in future.

BIBLIOGRAPHY

BIBLIOGRAPHY

PHP Tutorial -

<http://www.w3schools.com/php/default.asp>

<http://www.tutorialspoint.com/php/>

MySQL Tutorial -

<http://www.w3schools.com/php/php_mysql_intro.asp>

<http://www.tutorialspoint.com/mysql/index.htm>

JavaScript -

<http://www.w3schools.com/js/default.asp>

<http://www.tutorialspoint.com/javascript/index.htm>

CSS -

<http://www.w3schools.com/css/default.asp>

<http://www.tutorialspoint.com/css/index.htm>

HTML -

<http://www.w3schools.com/html/default.asp>

<http://www.tutorialspoint.com/html/index.htm>

HTML 5 - <http://www.tutorialspoint.com/html5/index.htm>

AJAX -

<http://www.w3schools.com/ajax/default.asp>

<http://www.tutorialspoint.com/ajax/index.htm>

**Question and answer site:**

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

Sams Teach Yourself PHP, MySQL and Apache All in One (5th Edition)  
Author: Julie Meloni

Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites

Author: Robin Nixon