

# NEPAL COLLEGE OF INFORMATION AND TECHNOLOGY

WEB-DESIGN COMPETITION

2018

---

## Election Portal

DISCOVER EVERYTHING ELECTION!

---

*Team Members:*

ASHISH TIWARI

NISCHAL LAL SHRESTHA

SAGAR DEVKOTA

July 16, 2018

logo.png

## **Abstract**

The purpose of Election Portal is to digitize election and election-related activities in Nepal. Election Portal offers a web application to store candidates information, election constituency details, latest election results. The problem statement relies on understanding a way to store all the datasets in the database, solving the complexity involved within and finding an interface to provide all these datasets to the visitors without a pain. Election Portal will provide opportunity to individual volunteers to update the election datasets, election result in real time. Each web page of the Election Portal will be equipped with either a search form or a filter form to enhance user's ability to get information very quickly.

## **1 Introduction**

Election Portal is a web application for individual visitors, journalists, students to get election results and other election-related activities at a single place. Election Portal will provide opportunity to individual volunteers to update the election datasets, election result in a real time.

## **2 Problem Statement**

The problem statement for Election Portal is based on understanding and developing a effective way to store all the datasets of election-related activities in a profound way, solving all the complexity involved with the datasets. It also relies on building a interface to filter and search the election datasets through the database in a effective manner.

## 3 UML Design

### 3.1 Use Case Diagram

election\_portal\_use\_case\_diagram.png

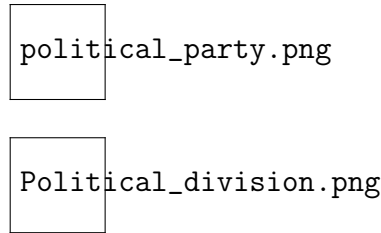
The Use Case Diagram of Election Portal contains the scope of the system. It lists all the functionalities offered by the system. We had identified following actors:

1. Visitor:- Visitor represents as the primary actor for this system. Visitors can browse web pages, view current results, search and filter candidates, constituencies, results.
2. Volunteer:- Volunteer act to update election results, add new datasets in the system. Volunteer needs to register first and then login to make changes in the system.
3. Super Admin:- Super Admin regulates the database. Super Admin approves new volunteer, remove volunteer and limit their access to the database.
4. Server:- A Server act to store the volunteers authentications information. Server acts to provide the web pages and datasets from database to users on their requests through a web browser.

Based on the above Use Case Diagram we can identify the following group of activities:

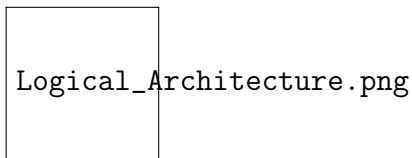
1. Visitors enter the web portal and browse various web pages.
2. Visitors can either search or filter the datasets through form.
3. Volunteer can Create, Read, Update and Delete datasets from the database.
4. Super Admin can approve new volunteer, limit access to volunteer and delete existing volunteer.

## 3.2 UML Class Diagram



# 4 Framework of the Development

## 4.1 Logical Architecture Diagram



There are three layers between User Interface and Database. Those are:

1. Data Access Layer(models.py)  
This layers defines everything about the data: how to access data, how we can validate data, what type of behaviors our data will show, and what type of relationship exists between different data.
2. Presentation Layer(Templates)  
Presentation layer will define how something should be displayed on a Web Page. Presentation layer takes information from the database and display it to the user in a Human-Understandable-Format. Presentation layer is completely free of any king of logic.
3. Business Logic Layer(views.py)  
Business Logic Layer is a bridge between models and templates. It provides appropriate way to supply specific templates requested by the data access layer.

## 5 Tools and Technology Used

1. Django:- Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source
2. U.S. Web Design System:- A design system for the federal government Design and build fast, accessible, mobile-friendly government websites backed by user research.
3. SQLite:- SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. SQLite is the most used database engine in the world.
4. JQuery:- JQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: Write less, do more. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.
5. Html and CSS:- HTML, HyperText Markup Language, gives content structure and meaning by defining that content as, for example, headings, paragraphs, or images. CSS, or Cascading Style Sheets, is a presentation language created to style the appearance of content—using, for example, fonts or colors.

## 6 Installation

### 6.1 Installing Python

Election Portal is a django app. Django is written in 100 percent pure Python code, so you'll need to install Python on your system. Django requires Python 2.3 or higher. If you're on Linux or Mac OS X, you probably already have Python installed.

### 6.2 Creating an isolated Python environment

A virtual environment (also called a virtualenv) is like a private box where we can stuff useful computer code into for a project we're working on. It is recommended that you use virtualenv to create isolated Python environments, so you can use different package versions for different projects, which is far more practical than installing Python packages system wide.

To install virtualenv:

```
pip install virtualenv
```

After you install virtualenv, create an isolated environment with the following command:

```
virtualenv myenvironment
```

Note: Here 'myenvironment' can be any valid name you like.

Now, You created a virtual environment let's activate it:

To Activate:

```
source myenvironment/bin/activate
```

### 6.3 Requirements.txt

This file consists all the dependencies of this app.

Our web app needs following packages to work correctly:

```
chardet==3.0.4
```

```
Django==2.0.7
```

```
doc8==0.8.0
```

```
docutils==0.14
```

```
pbr==4.0.4
```

```
pkg-resources==0.0.0
pytz==2018.5
restructuredtext-lint==1.1.3
six==1.11.0
stevedore==1.28.0
```

To install all the dependencies, just type:

```
pip install -r Requirements.txt
```

Make Sure You have an active internet connection.

Now, All our dependencies are fulfilled.

In your terminal, Change Directory to the Base Directory of our website.  
i.e the folder which consist manage.py:

```
cd ElectionPortalWebapp
```

Then:

```
python manage.py collectstatic
python manage.py makemigrations
python manage.py migrate
```

To Run The Server:

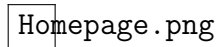
```
python manage.py runserver
```

Our web app is ready for local development.

## 7 Graphical User Interface of the System

Election Portal has a GUI so that visitors can easily browse the portal with the least effort. Every page has a header and footer. The header consists of the Navigation menu so that visitors can navigate throughout the webpage without being lost.

### 7.1 Homepage


Homepage.png

Beside Header and Footer, Homepage of Election Portal has a search form which allows visitors to search election datasets without a pain. The homepage has a map, so any users can filter the datasets via district.




## 7.2 List of Political Party


All the political parties registered will be listed here along with their party image. Given figure shows same image because we don't have update the database with real datas.

 List\_of\_Political\_Party.png


### 7.3 Details of Political Party

 Details\_of\_Political\_Party.png


### 7.4 Details of Nominee

 Details\_of\_a\_nominee.png


### 7.5 Details of Constituency

 Details\_of\_Constituency.png


### 7.6 Form to Add Nominee

 Form\_To\_Add\_Candidate.png

### 7.7 Form to Add Political Party

 Form\_To\_Add\_Political\_Party.png

### 7.8 Form To Add Province

 Form\_To\_Add\_Province.png

## References

- [1] Wikipedia - Use Case Diagram  
<https://en.wikipedia.org/wiki/Use-case-diagram>
- [2] Wikipedia - Class Diagram  
<https://en.wikipedia.org/wiki/Class-diagram>
- [3] Fedrous - Web Design Report  
<https://slideshare.net/ferdous/web-design-project-report>
- [4] Django Documentation;  
<https://docs.djangoproject.com/en/2.0/>