Karthik M (15CO221), Tarun Karthik (15CO155)

National INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

DATABASE MANAGEMENT SYSTEMS PROJECT ON MUNICPAL e-GOVERNANCE

Table of Contents

[About 2](#_Toc497595187)

[Necessity 2](#_Toc497595188)

[Objectives 2](#_Toc497595189)

[Requirements Specification 3](#_Toc497595190)

[Conceptual Schema (ER Diagram) 4](#_Toc497595191)

[Relational Schema 6](#_Toc497595192)

[Normalization of the relational schema 6](#_Toc497595193)

[Specific Functional Requirements 6](#_Toc497595194)

[Database Administrator 7](#_Toc497595195)

[Public 7](#_Toc497595196)

[Elected Representative 15](#_Toc497595197)

[HOD of a department 16](#_Toc497595198)

[Contractor of a public work 23](#_Toc497595199)

[Tools Used to implement the database 26](#_Toc497595200)

[Code Samples used in the project 27](#_Toc497595201)

[HTML, CSS 27](#_Toc497595202)

[PHP 29](#_Toc497595203)

[GitHub link to the PHP codes 32](#_Toc497595204)

[Conclusion and Future Work 32](#_Toc497595205)

**Municipal e-Governance**

**Database Management System Project**

# About

e-Governance is a form of public administration making use of information and communication technologies to enhance the access and delivery of government services to benefit citizens, employees and management of urban local bodies. It aims to help strengthen government’s drive toward effective governance and increase transparency to better manage social and economic resources for development.

# Necessity

* Manual recording of data is time consuming and prone to error
* To avoid manual filing of application for trade, birth & death registration, etc.
* Huge potential for data manipulation and human errors.
* No centralized database in place.
* Search and retrieval of data becomes difficult.

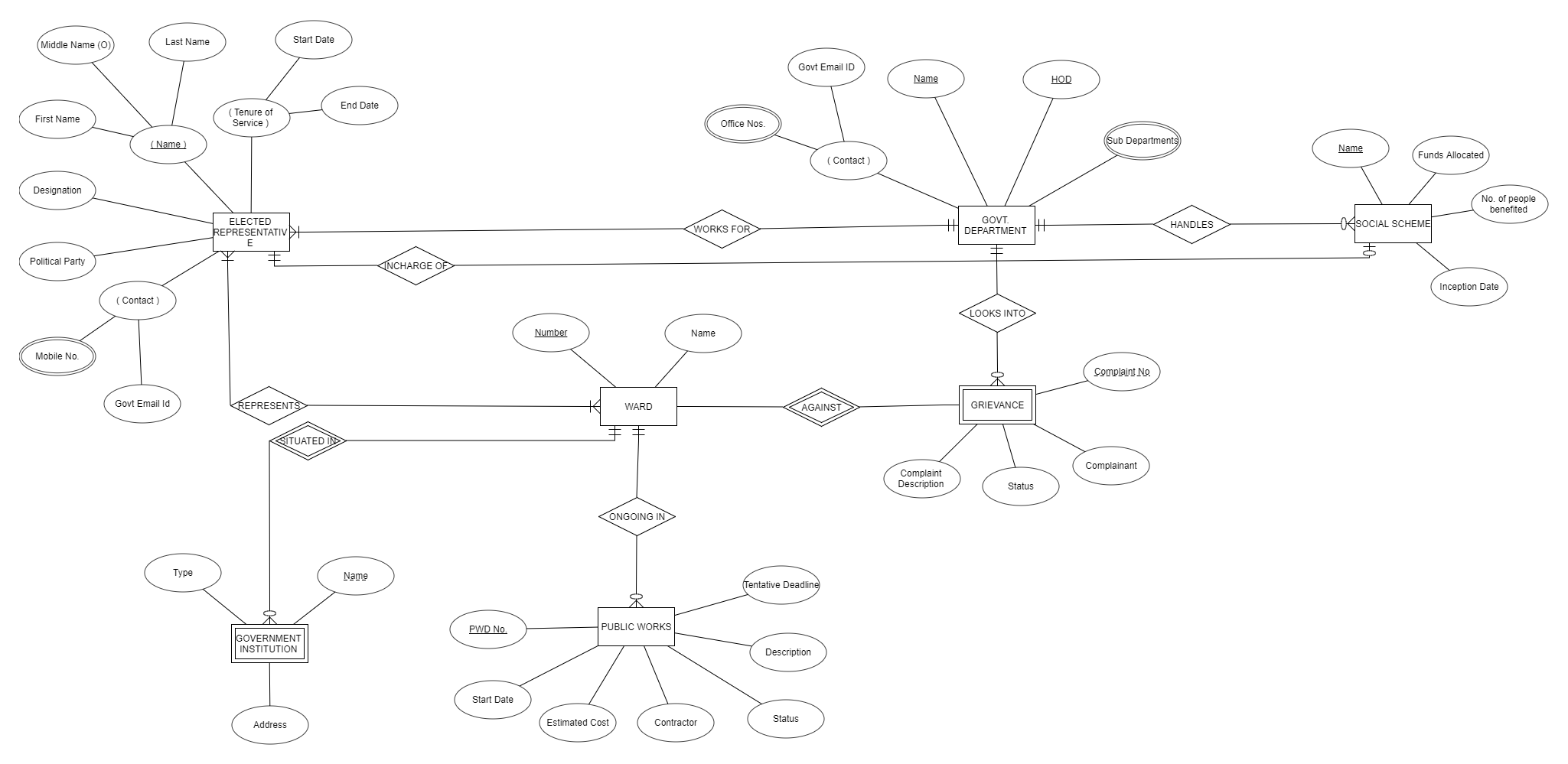
# Objectives

* Promote people centric administration
* Changing from age-old methods of book and pen to computer based governance
* Improving quality of city governance
* Systematically improve administration in terms of record keeping, searching, storing and processing records
* Track the progress of various wards based on metrics
* Provide a website for the corporation/municipality
* Improve accountability of government towards citizens

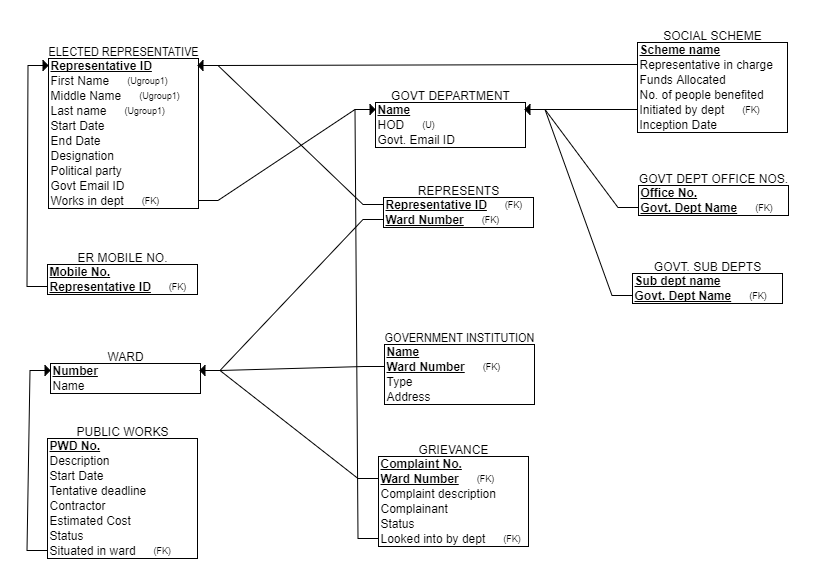
# Requirements Specification

* A municipality has many wards, each identified by a ward number and a ward name.
* Every elected representative of the municipality has a first name, an optional middle name and last name. He/she has a designation (Corporator/MLA/MP) and is affiliated to a political party (BJP/Congress). He/she has a fixed term marked by a start date and an end date. An elected representative can have many wards under his/her jurisdiction (e.g.: A corporator has exactly one ward, an MLA has more than one ward and an MP has many wards under his jurisdiction). Thereby, a ward can have more than one elected representative in-charge of it. Every elected representative has an official e-mail ID provided by the government and may possess multiple mobile numbers
* A municipal corporation has many departments. Every govt. department has its own HOD and a department name. Each department can have further sub-departments (e.g.: The Health department can have sub-departments like Food Licenses Unit, Registration of Birth and Death, Communicable Disease Control, etc.). Every department is given an official govt. Email ID and can have multiple office numbers. More than 1 elected representative can work for a govt. Department but every govt. department should have at-least one elected representative
* There can be many social schemes run by the government in the municipality. Each scheme has a unique name and an inception date. The database should keep track of the funds allocated and the number of people benefited by the scheme. Each scheme should have one elected representative in-charge of it.
* The municipal corporation can run many government institutions like schools, colleges, hospitals. A ward can have many govt. Institutions and each Government institution can be uniquely identified by its name and the ward that it is situated in.
* Also, there can be many public works going in each ward (e.g.: roads, railways, bridges, parks, electrical grid, etc.). Each public work is identified by a unique PWD No. and has a tentative deadline set by the contractor, an estimated cost and a qualitative status (not started, ongoing, nearing completion, completed)
* The public can voice their grievances which will be looked into by a govt. Department. Each grievance will be uniquely identified in a ward by a complaint number. The complainant name and the status (grievance redressed or not) are also to be recorded.

# Conceptual Schema (ER Diagram)



# Relational Schema



# Normalization of the relational schema

The mapping of the conceptual schema to the relation schema produced relations where every functional dependency X -> Y of relation R was such that X is a primary key of R. In other words, the relational schema is already BCNF normalized without any non-trivial functional dependencies originating from non-prime attributes of the relational schema.

# Specific Functional Requirements

A tentative list of database users and their interactions with the database in the form of input, processing, output has been presented here.

## Database Administrator

1. Add details of an elected representative

*Input:* Elected representative name (first name, middle name, last name), start date and end date of service, designation, political party, contact (mobile no., govt. email-ID)

*Processing:* Add details of representative into the database

1. Add wards

*Input:* Ward name, ward no.

*Processing:* Add details of ward to the database

1. Add government departments

*Input:* Department name, department HOD

*Processing:* Add details of govt. department to the database

## Public

1. Submit a grievance

*Input:* Complaint description, complainant name, (complaint against) ward number

*Processing:* Generate a complaint number, set “Status” as NULL and store the submitted details in the database

1. View status of a grievance

*Input:* Ward no., complaint no.

*Processing:* Fetch details regarding the complaint

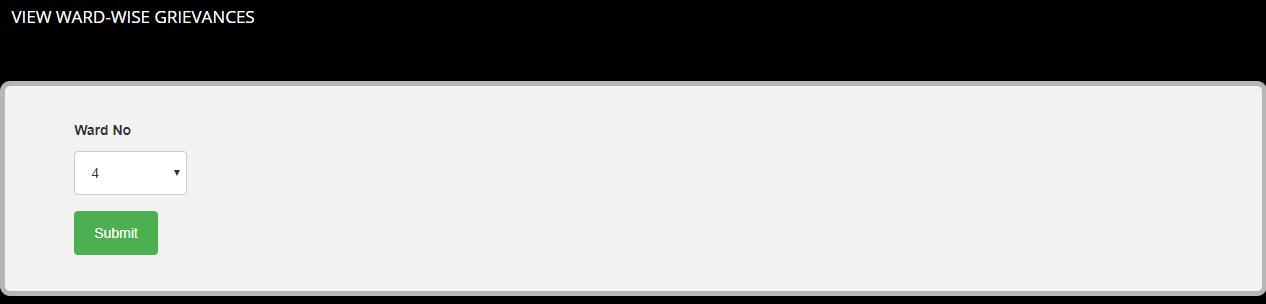
*Output:* Display complaint details

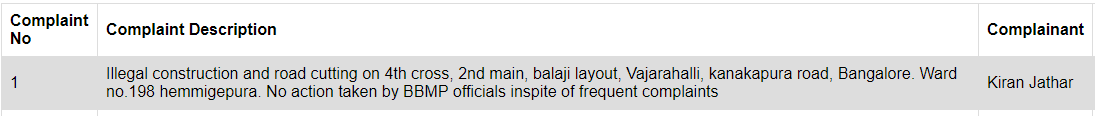
1. View ward-wise details of grievances

*Input:* Ward no.

*Processing:* Select all grievances with the given ward no.

*Output:* Display the list of grievances with the given ward no.





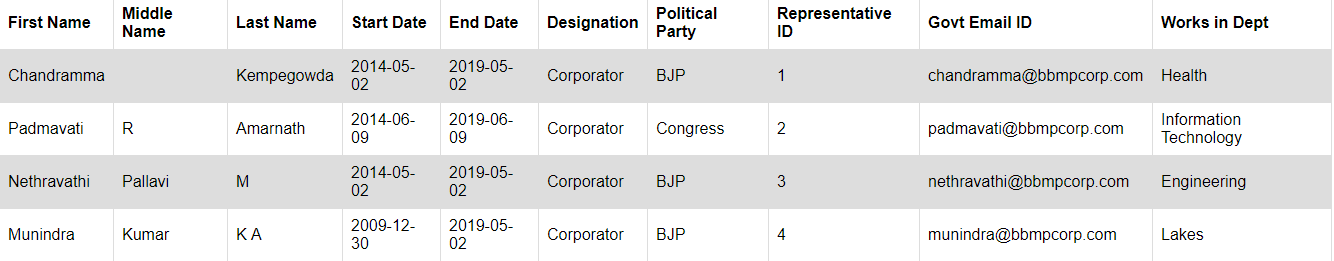
1. View list of all elected representatives in the municipality

*Input:* Select “View all representatives”

*Processing:* Retrieve details of all elected representatives

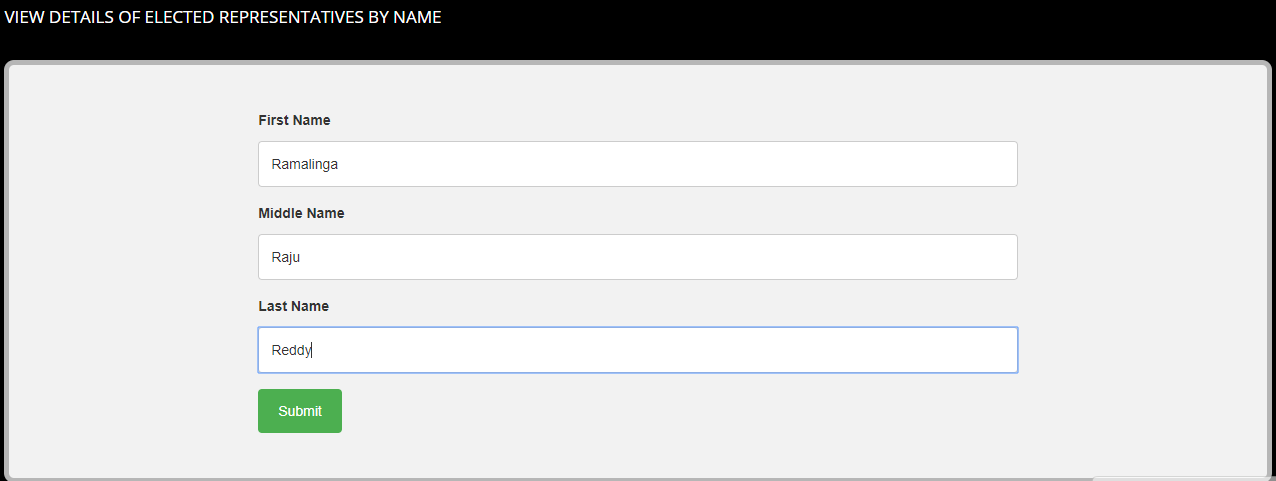
*Output:* Display all representatives of the municipality

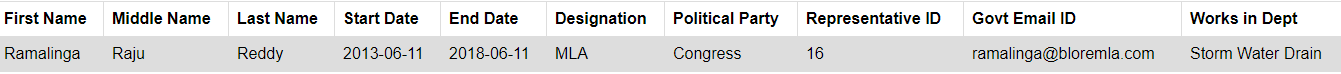




1. View details of an elected representative by name

*Input:* Name of representative

*Output:* Details of the elected representative

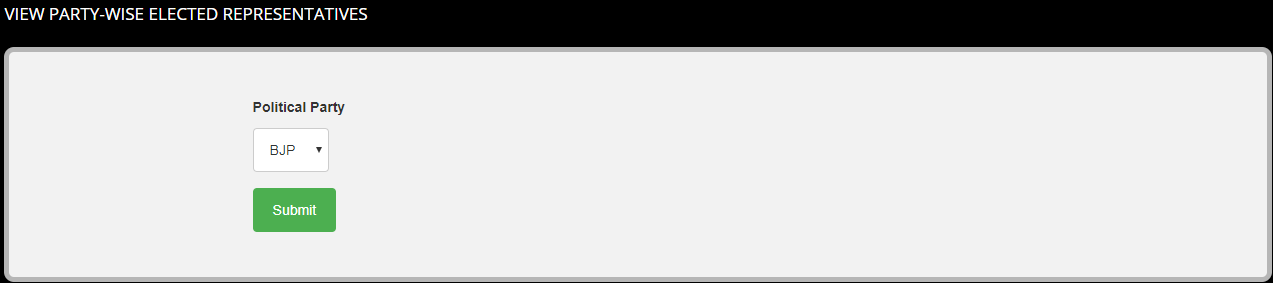


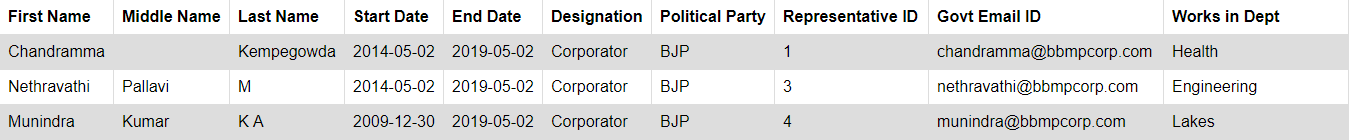
1. List all elected representatives by their political party

*Input:* Political party

*Processing:* Select all representatives affiliated to the given political party

*Output:* Display the list of representatives



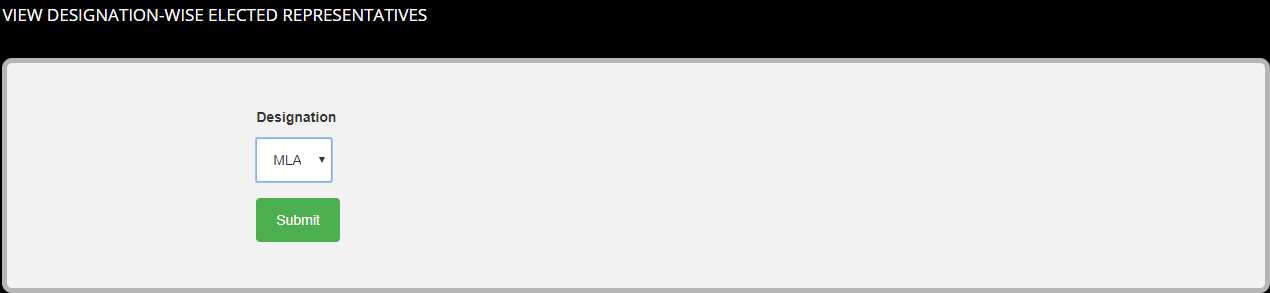


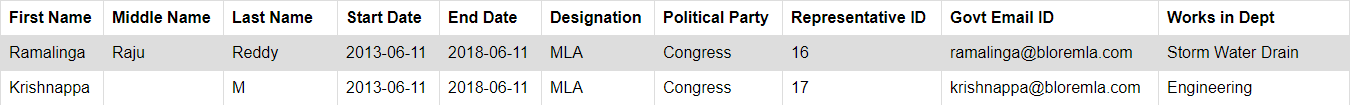
1. List all corporators, MLAs or MPs and the wards under their jurisdiction

*Input:* Designation (Corporator, MLA, MP)

*Processing:* Fetch details of all representatives with the input designation

*Output:* Display all representatives with the given designation

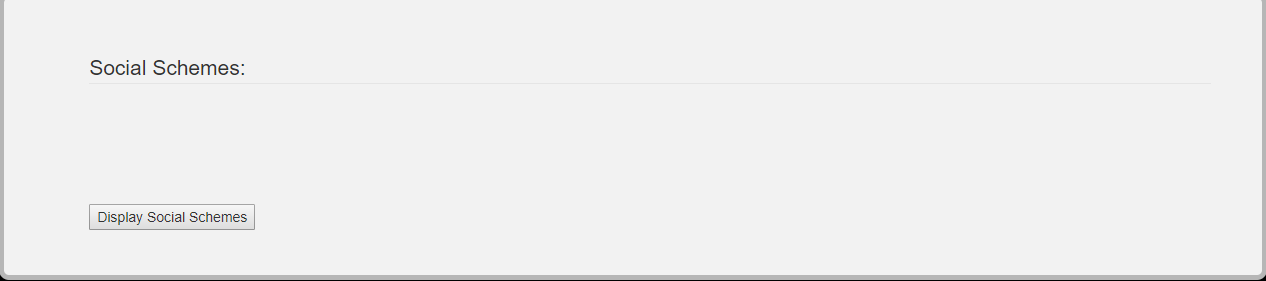


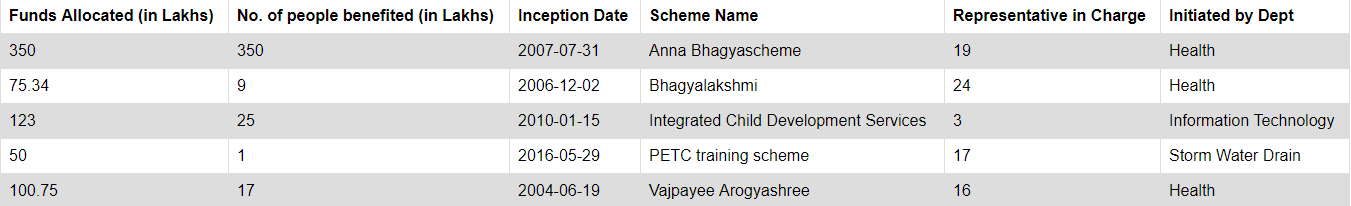


1. Check the details of various social schemes in the municipality

*Input:* Select “View All Schemes”

*Output:* Display all the social schemes in the municipality

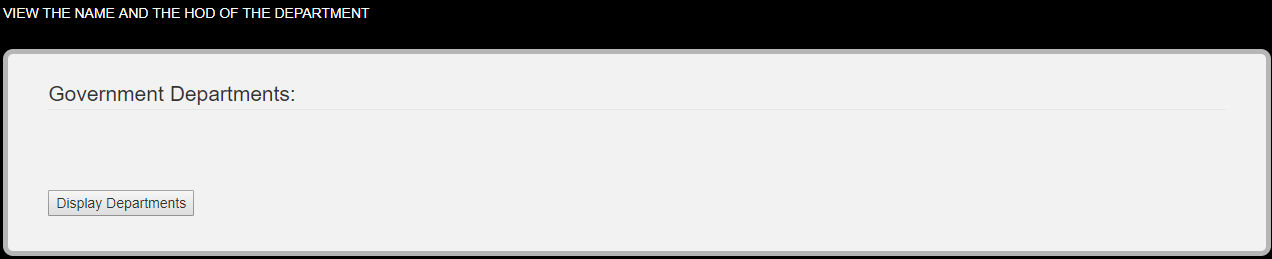


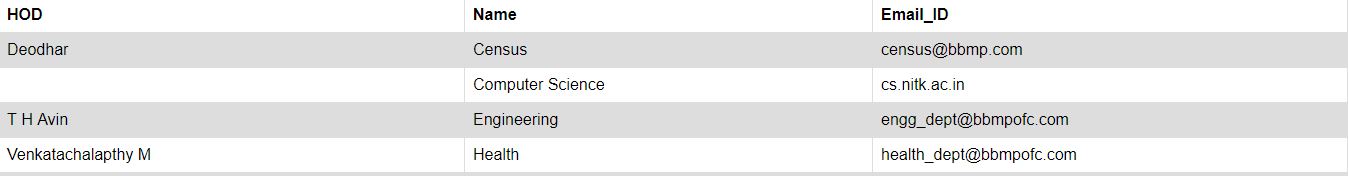


1. View details of all government departments

*Input:* Select “View All Govt. Departments”

*Output:* Display all the Govt. Departments



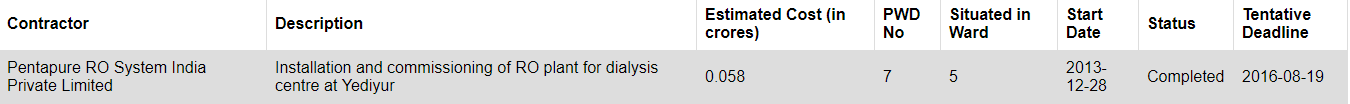


1. View ward-wise details of public works

*Input:* Ward no.

*Processing:* Fetch details of all public works with the given ward no.

*Output:* Display all the ongoing public works in the ward

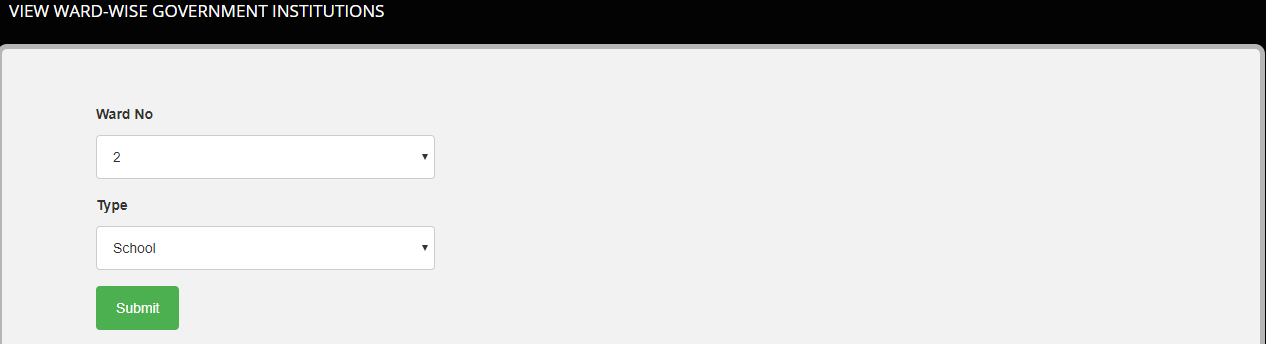


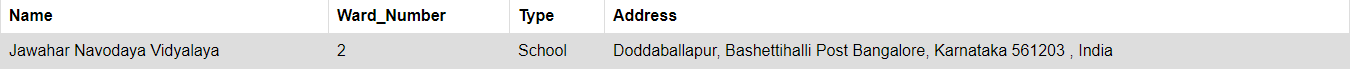
1. View ward-wise details of a government institution type

*Input:* Ward no., type

*Processing:* Fetch details of all the government institutions with the given ward no. and type

*Output:* Display all the requested government institutions in the ward

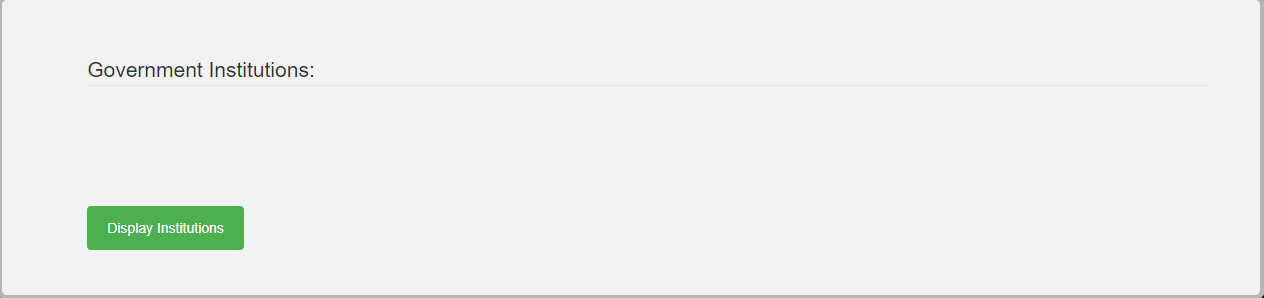


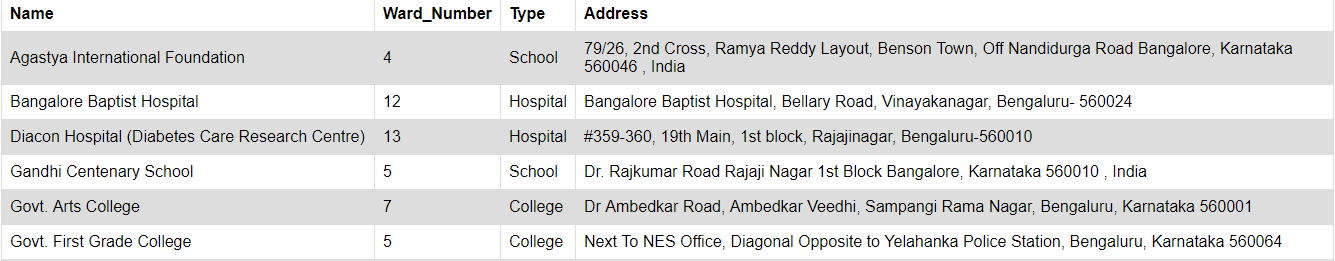


1. View all government institution

*Input:* Select “View All Govt. institution”

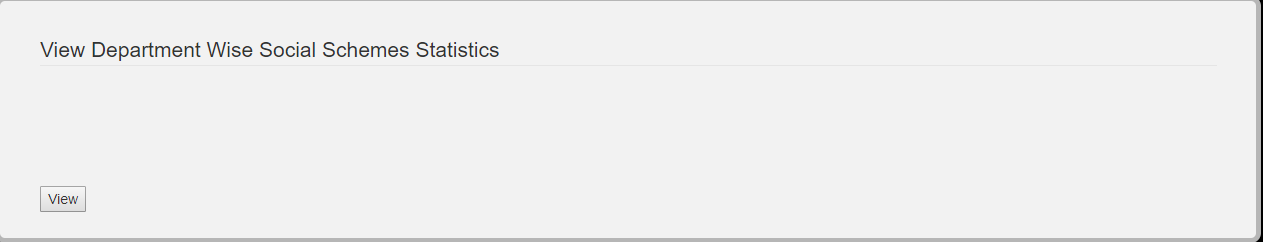
*Output:* Display all the Govt. institution





1. View total funds allocated and total number of people benefited for each government department

*Input:* Select “View Department Wise Social Schemes Statistics”

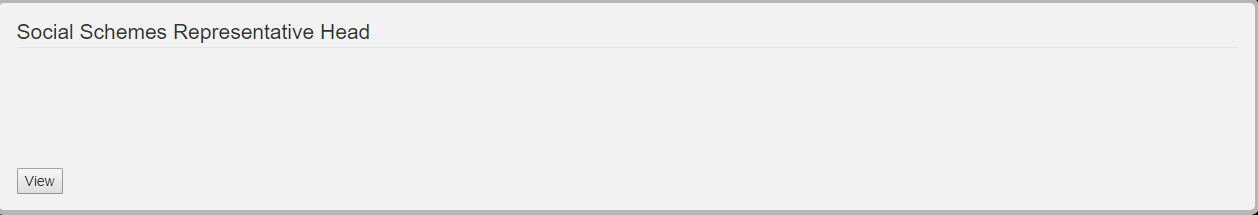
*Output:* Display the department wise social schemes statistics

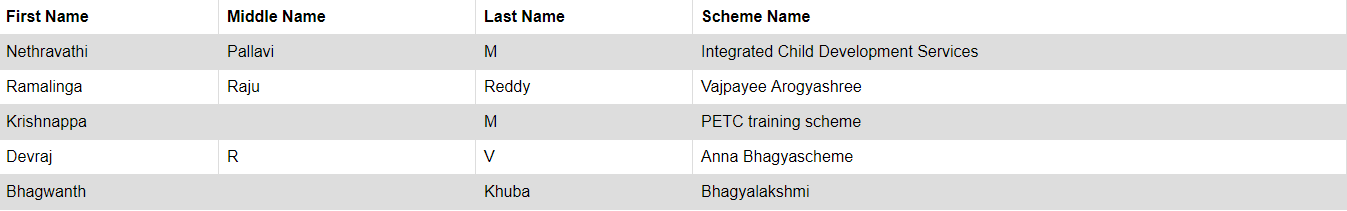


1. View a list of social schemes and the elected representatives in charge of them

*Input:* Select “View Social Schemes Representative Head”

*Output:* Display the corresponding list

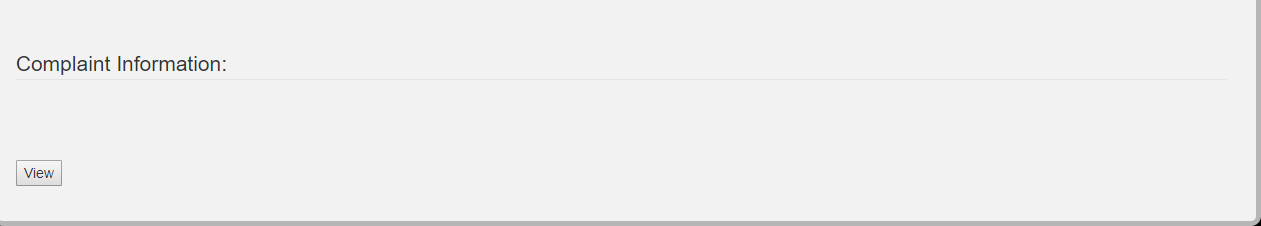


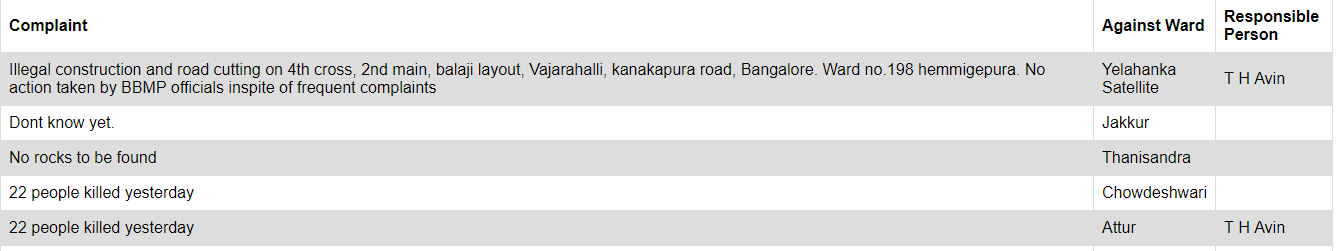


1. View all grievances, wards against which the grievance is made and the person (government department HOD) responsible for it

*Input:* Select “View complaint information”

*Output:* Display the corresponding list



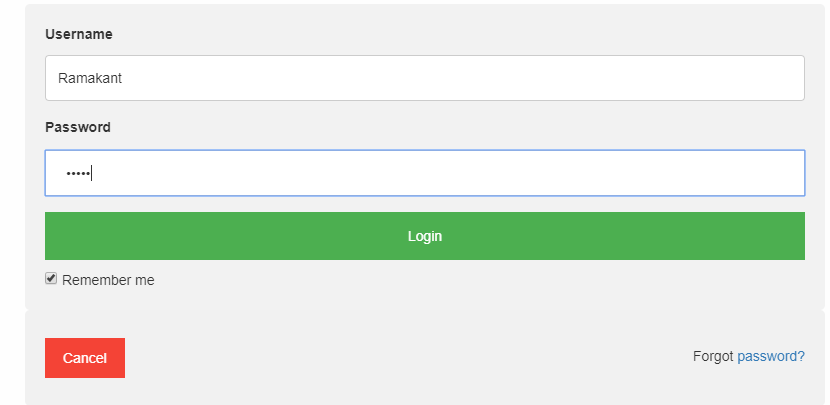


## Elected Representative

1. Login to the ER page

*Input:* Username, Password

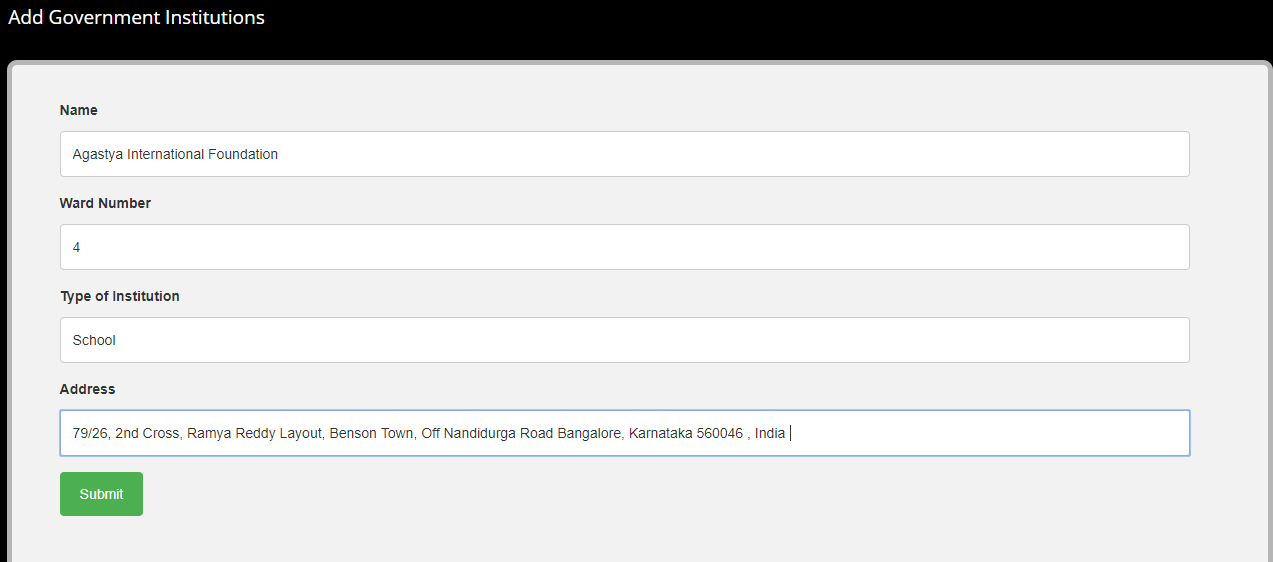
*Processing:* Validation of username, password

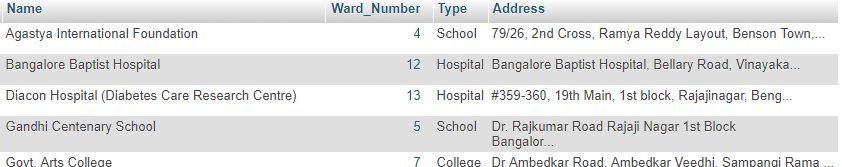


1. Add government institutions situated in his/her ward

*Input:* Ward no., institution name, type, address

*Processing:* Add details of the institution to the database





1. Add public works situated in his/her ward

*Input:* PWD No., Start date, contractor, description

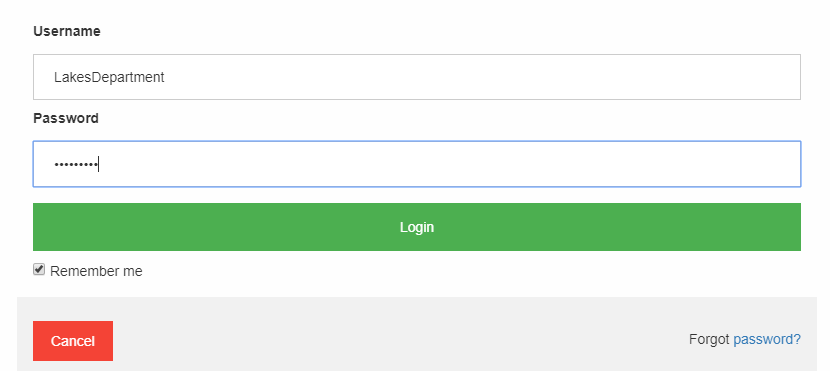
*Processing:* Add details of the public work to the database, set “Estimated Cost”, “Status” and “Tentative deadline” to NULL

## HOD of a department

1. Login to the HOD page

*Input:* Username, Password (Common to HODs of all government departments)

*Processing:* Validation of username, password

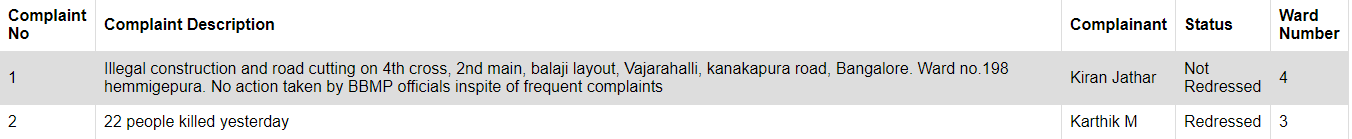
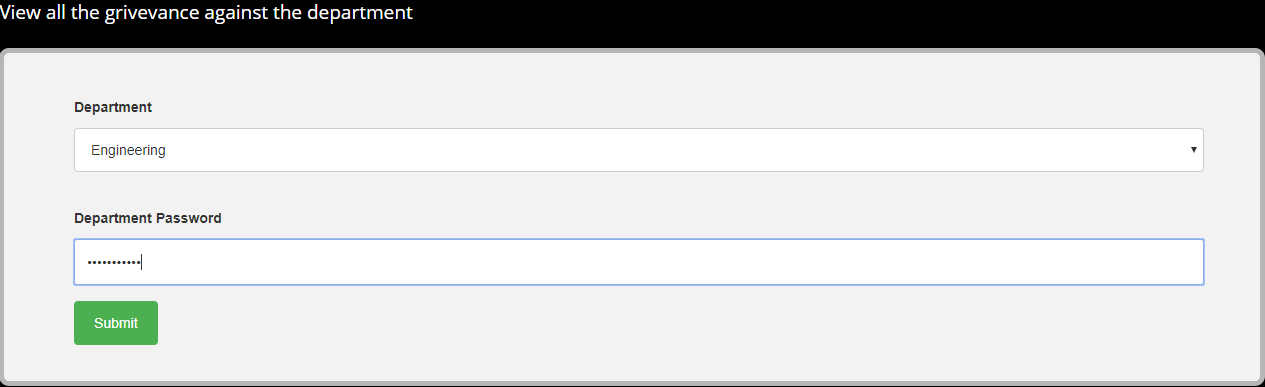


1. List all grievances against the department

*Input:* Select “View Pending Grievances”

*Processing:* Select all grievances not yet addressed

*Output:* Show all grievances chosen for redressal by the department

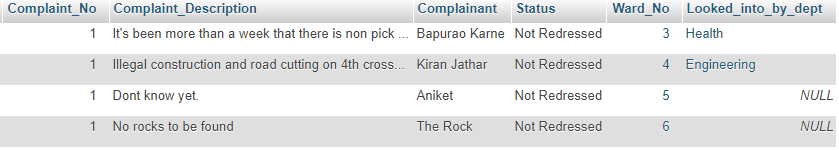


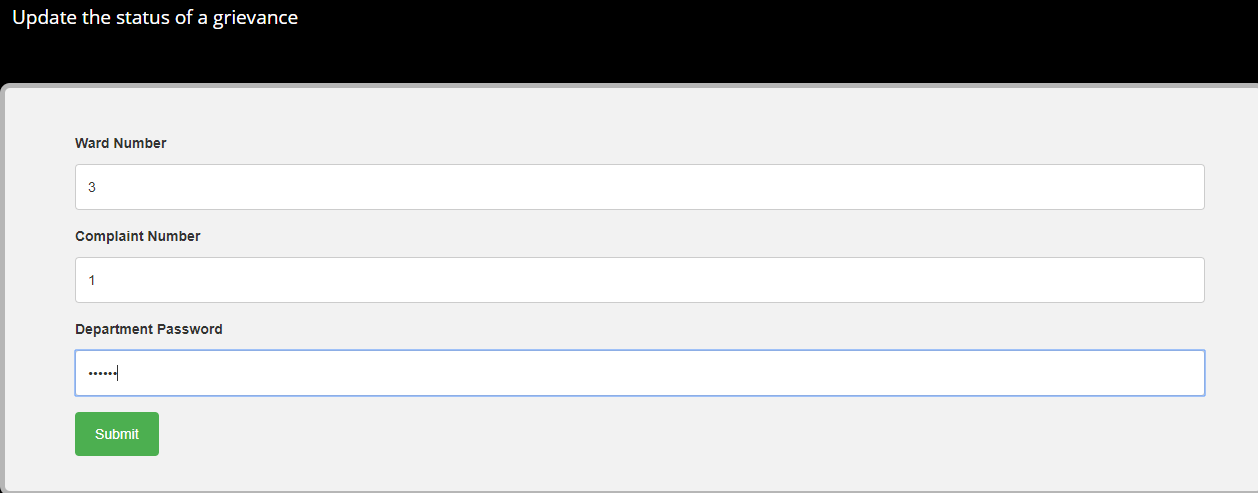
1. Update the status of a grievance

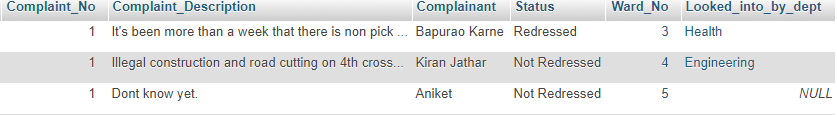
*Input:* Complaint No, new status

*Processing:* Update status of the grievance in the database

*Output:* Error message, if the grievance status was not set as “Redressed”



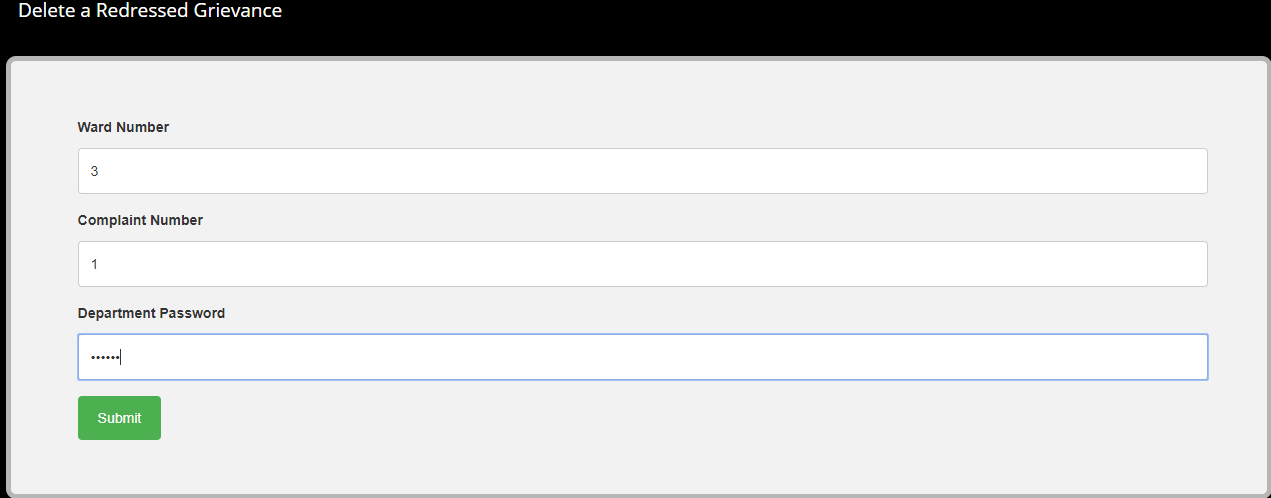




1. Delete a grievance after it has been redressed

*Input:* Complaint No., “delete” option

*Processing:* Delete a complaint only if the status is set as “Redressed”

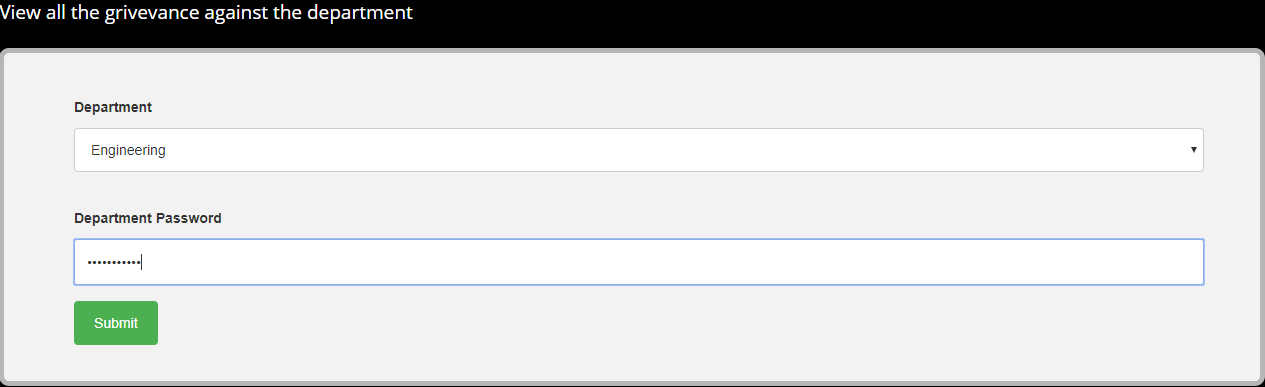


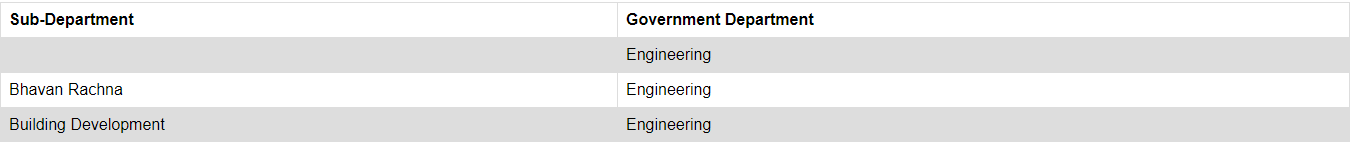


1. View all sub-departments under a department

*Input:* Department name

*Output:* View all sub-departments under the given department name

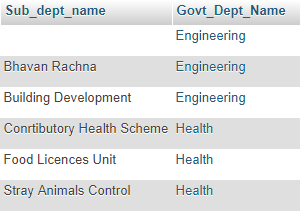


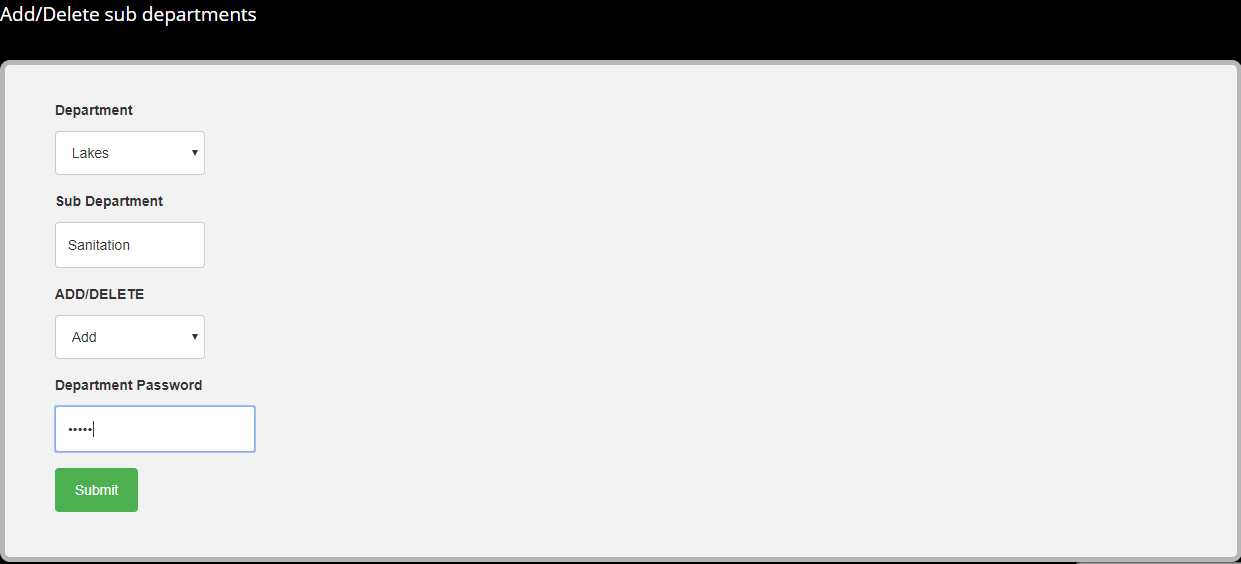


1. Add/Delete government sub departments

*Input:* Select “Add Sub-department” or “Delete Sub-department”, sub-department name

*Processing:* Update the database accordingly



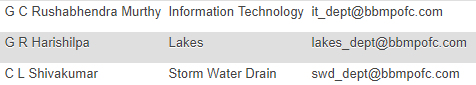


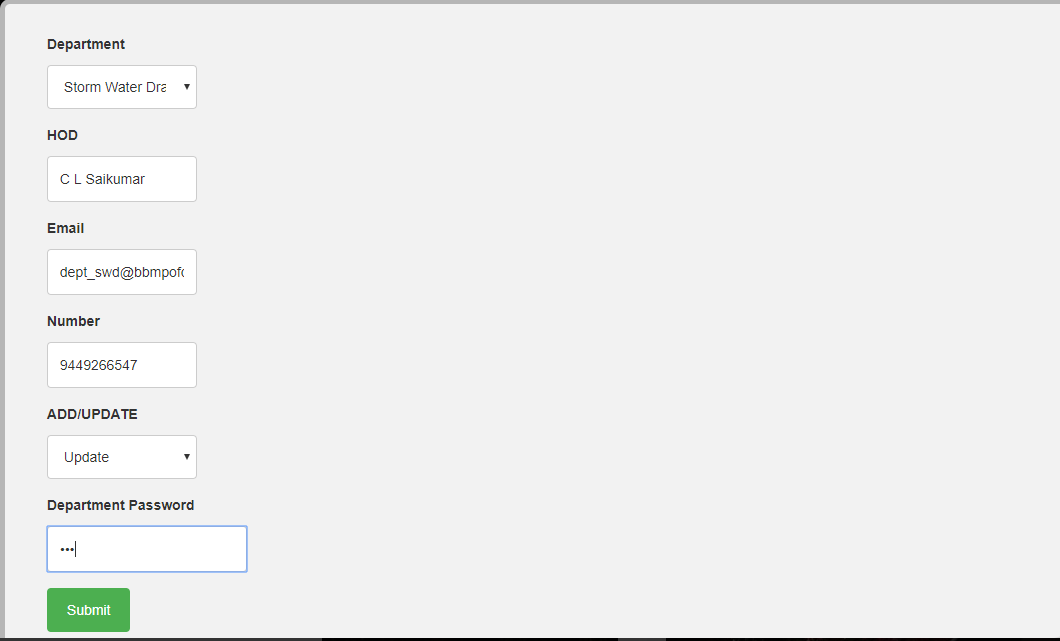


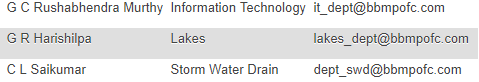
1. Add/Update contacts of the department

*Input:* Select “Add” or “Update contacts” respectively. Fill in a new govt. email ID and an office contact number.

*Processing:* Update contact details in the database



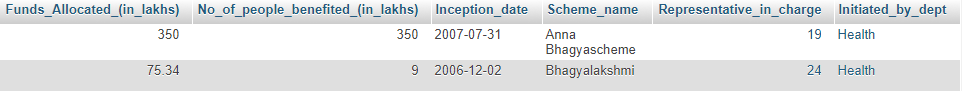


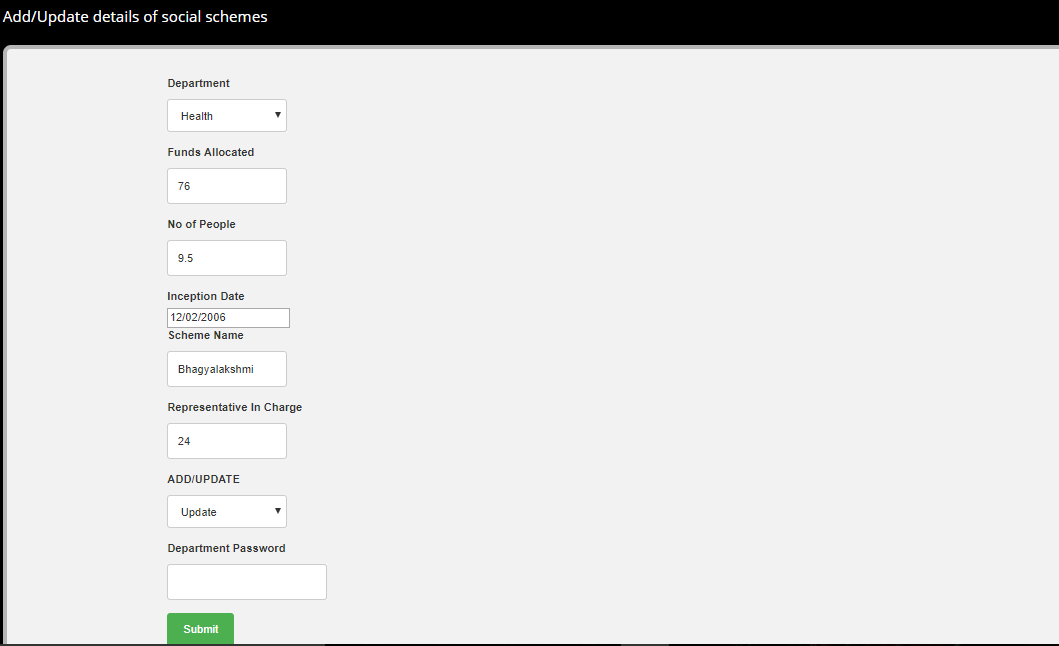


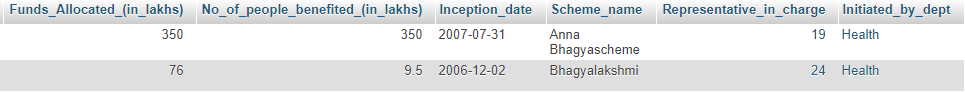
1. Add/Update details of social schemes for which the department is in charge

*Input:* Name of social scheme, funds allocated, inception date, elected representative in-charge

*Processing:* Add the new social scheme to the database, set “No. of people benefited” as 0



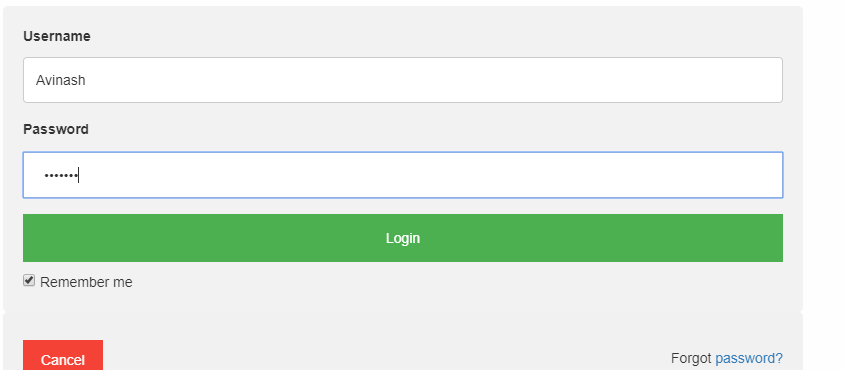




## Contractor of a public work

1. Login to the system

*Input:* Username, Password

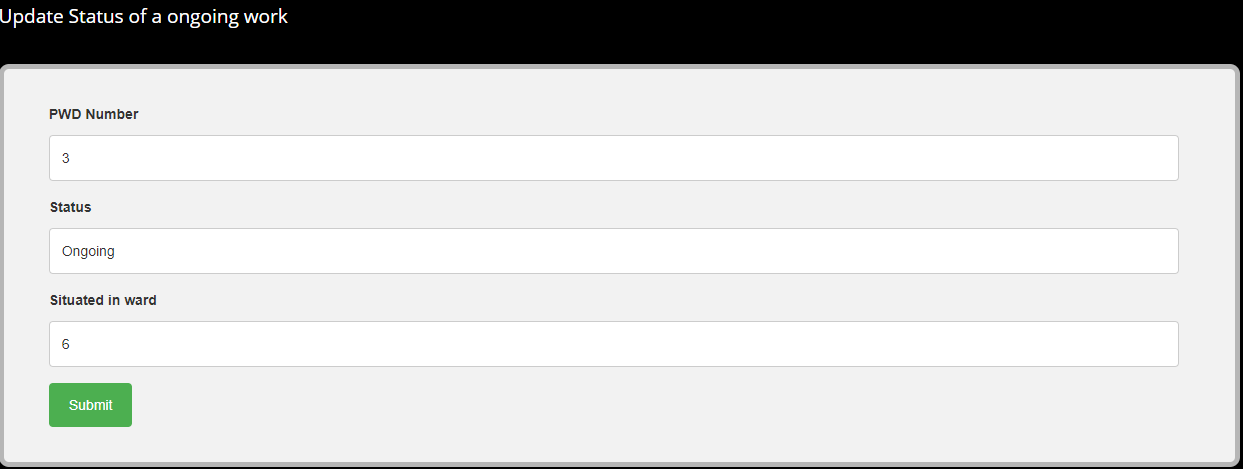
*Processing:* Validation of username, password

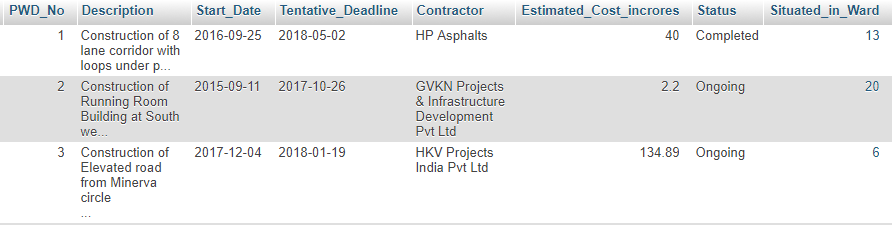
1. Update the status of the ongoing work

*Input:* PWD No., status

*Processing:* Update the database



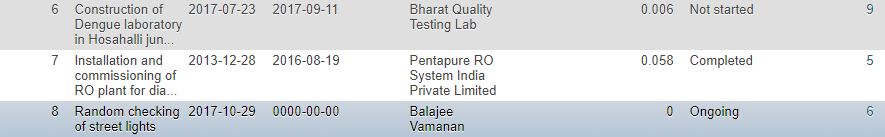




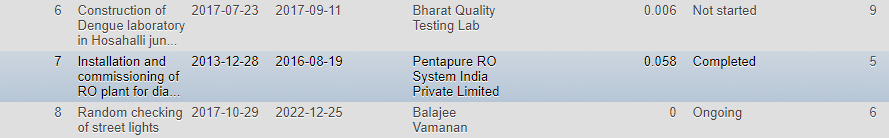
1. Set deadlines for the public work

*Input:* PWD No., deadline (date)

*Processing:* Update the database



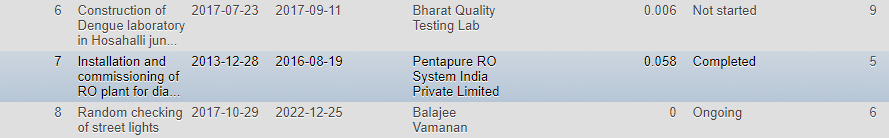


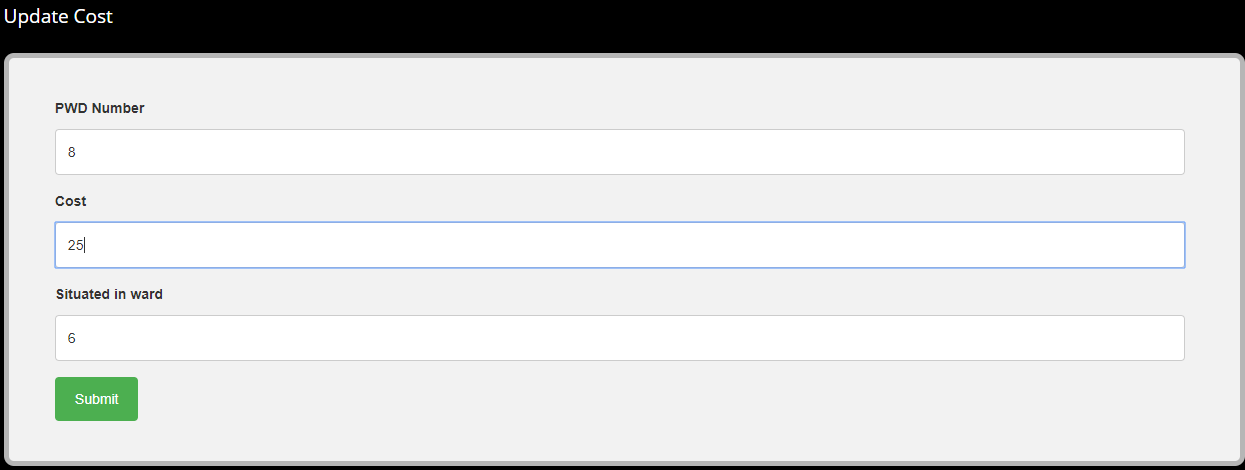


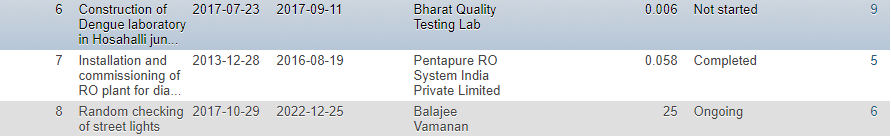
1. Update estimated cost

*Input:* PWD No., estimated cost

*Processing:* Update the database







# Tools Used to implement the database

The web application is hosted on a webhost by the name 000webhost (<https://in.000webhost.com/>). The tools used are:

* HTML, CSS, JS for the front end
* PHP, MySQL for the back-end
* PhpMyAdmin is used to manage the database.

The webhost provides access to phpMyAdmin from within the website

# Code Samples used in the project

## HTML, CSS

<**!DOCTYPE html>**

**<html>**

**<head>**

**<style>**

**input[**type=password**],** select**,** textarea **{**

**width: 200px;**

**padding: 12px;**

**border: 1px solid #ccc;**

**border-radius: 4px;**

**box-sizing: border-box;**

**margin-top: 6px;**

**margin-bottom: 16px;**

**resize: vertical;**

**}**

input**[**type=text**],** select**,** textarea **{**

**width: 150px;**

**padding: 12px;**

**border: 1px solid #ccc;**

**border-radius: 4px;**

**box-sizing: border-box;**

**margin-top: 6px;**

**margin-bottom: 16px;**

**resize: vertical;**

**}**

input**[**type=submit**]** **{**

**background-color: #4CAF50;**

**color: white;**

**padding: 12px 20px;**

**border: none;**

**border-radius: 4px;**

**cursor: pointer;**

**}**

input**[**type=submit**]:hover** **{**

**background-color: #45a049;**

**}**

**.**container **{**

**border-radius: 5px;**

**background-color: #f2f2f2;**

**padding: 20px;**

**}**

</style**>**

</head**>**

<body**>**

<div class**="container">**

<form action**="/Government Departments/HOD/addupdatecontact.**php" method**="post">**

<label for**="Ward No">**Department</label**>**<br**>**

<select name**="dept" id="Ward No" width="200px">**

<option **>**Engineering</option**>**

<option**>**Health</option**>**

<option**>**Information Technology</option**>**

<option**>**Lakes</option**>**

<option**>**Storm Water Drain</option**>**

</select**>**<br**>**

<label for**="HOD">**HOD</label**>**<br**>**

<input type**="text" id="HOD" name="HOD" width="100px" required>**

<br**>**

<label for**="email">**Email</label**>**<br**>**

<input type**="text" id="email" name="email" width="100px" required>**

<br**>**

<label for**="no">**Number</label**>**<br**>**

<input type**="text" id="no" name="number" width="100px" required>**

<br**>**

<label for**="Ward">**ADD/UPDATE</label**>**<br**>**

<select name**="adddelete" id="Ward" width="200px">**

<option **>**Add</option**>**

<option**>**Update</option**>**

</select**>**

<br**>**

<label for**="subjec">**Department Password</label**>**<br**>**

<input type**="password" id="subjec" name="password" width="100px" required>**

<br**>**

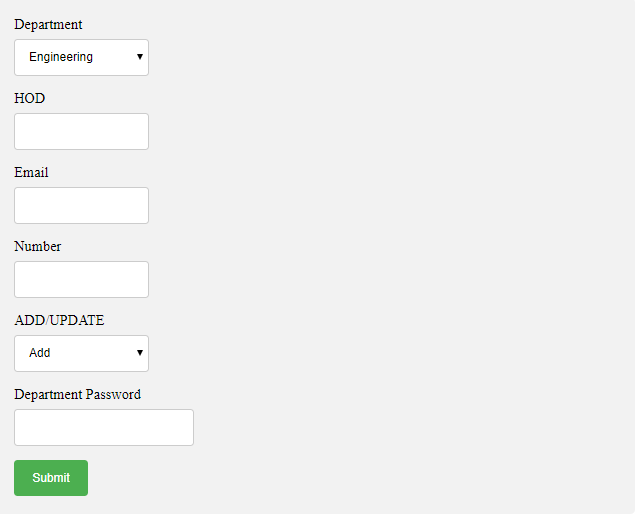
<input type**="submit" value="Submit" name="submit">**

</form**>**

</div**>**

</body**>**

</html**>**



## PHP

<?php

$servername = "localhost";

$username = "id2963861\_wp\_2be0db2d9519ee3205bea03d32c84df8";

$password = "tarunkarthik";

$dbname = "id2963861\_wp\_2be0db2d9519ee3205bea03d32c84df8";

$conn = **mysqli\_connect**($servername, $username, $password, $dbname);

**if** ($conn->connect\_error) {

**die**("Connection failed: " . $conn->connect\_error);

}

**if**($\_POST)

{

$dept = $\_POST['dept'];

$email = $\_POST['email'];

$number = $\_POST['number'];

$addupdate = $\_POST['adddelete'];

$password = $\_POST['password'];

$hod = $\_POST['HOD'];

$sql = "SELECT \* FROM `department\_login` WHERE password = '**$password**'";

$result = **mysqli\_query**($conn,$sql);

$count = **mysqli\_num\_rows**($result);

**if**($count!=1)

{

**echo** '<script type="text/javascript">alert("Wrong Password");</script>';

}

**else**

{

**if**($addupdate=="Add")

{

$sql2 = "SELECT \* FROM `Government\_Department` WHERE Name='**$dept**'";

$result2 = **mysqli\_query**($conn,$sql2);

$count2 = **mysqli\_num\_rows**($result2);

**if**($count2!=1)

{

$sql = "INSERT INTO `Government\_Department`(`HOD`,`Name`, `Govt\_Dept\_Name`) VALUES ('**$hod**','**$dept**','**$email**')";

$result4 = **mysqli\_query**($conn,$sql);

**echo** '<script type="text/javascript">alert("record added");</script>';

}

**else**

{

**echo** '<script type="text/javascript">alert("record already exists");</script>';

}

$sql3 = "SELECT \* FROM `Govt\_Dept\_Office\_Nos` WHERE Govt\_Dept\_Name='**$dept**'";

$result3 = **mysqli\_query**($conn,$sql3);

$count3 = **mysqli\_num\_rows**($result3);

**if**($count3!=1)

{

$sql = "INSERT INTO `Govt\_Dept\_Office\_Nos`(`OFfice\_No`,`Govt\_Dept\_Name`) VALUES ('**$number**','**$dept**')";

$result4 = **mysqli\_query**($conn,$sql);

**echo** '<script type="text/javascript">alert("record added");</script>';

}

**else**

{

**echo** '<script type="text/javascript">alert("record already exists");</script>';

}

}

**else**

{

$sql2 = "SELECT \* FROM `Government\_Department` WHERE Name='**$dept**'";

$result2 = **mysqli\_query**($conn,$sql2);

$count2 = **mysqli\_num\_rows**($result2);

**if**($count2==1)

{

$sql = "UPDATE `Government\_Department` SET `HOD`='**$hod**',`Email\_ID`='**$email**' WHERE `Name`='**$dept**'";

$result4 = **mysqli\_query**($conn,$sql);

**echo** '<script type="text/javascript">alert("record added");</script>';

}

**else**

{

**echo** '<script type="text/javascript">alert("record already exists");</script>';

}

$sql3 = "SELECT \* FROM `Govt\_Dept\_Office\_Nos` WHERE Govt\_Dept\_Name='**$dept**'";

$result3 = **mysqli\_query**($conn,$sql3);

$count3 = **mysqli\_num\_rows**($result3);

**if**($count3==1)

{

$sql = "UPDATE `Govt\_Dept\_Office\_Nos` SET `Office\_No`='**$number**' WHERE Govt\_Dept\_Name='**$dept**'";

$result4 = **mysqli\_query**($conn,$sql);

**echo** '<script type="text/javascript">alert("record updated");</script>';

}

**else**

{

**echo** '<script type="text/javascript">alert("record does not exists");</script>';

}

}

}

}

?>

# GitHub link to the PHP codes

<https://github.com/Tarun-Karthik/Municipal-E-Governence-Website>

# Conclusion and Future Work

The tools that we used are the website 000webhost that helped us get a domain and a server and allowed us to host the website. All the functionalities have been met according to the project report. The new languages that we had to learn are

* PHP
* HTML
* CSS
* JavaScript.

One way to possibly expand the website is to a point such that it can be actually used by the public/government to provide a better, faster and convenient governance for the public.