# A PROJECT REPORT ON

**Online Leave Submission System**

Submitted in partial fulfillment of the requirements

For the award of the degree of

**BACHELOR OF TECHNOLOGY**

In

Computer Science & Engineering

By

**A. Lavanya 411503**

**D. Ramya sree 411520**

**K. Chelsea Jyotsna 411534**

Under the esteemed guidance of

**Ms G.JAYA LAKSHMI**

**Assistant Professor, CSE Department**



**Department of Computer Science and Engineering**

**NATIONAL INSTITUTE OF TECHNOLOGY,**

**Andhra Pradesh.**

**ACKNOWLEDGEMENT**

I express my deep gratitude and regards to **Smt. G. Jaya Lakshmi**, Internal Guide and Adhoc faculty, Department of Computer Science & Engineering for her encouragement and valuable guidance in bringing shape to this dissertation.

I am thankful to all the Professors and Faculty Members in the department for their teachings and academic support and thanks to Technical Staff and Non-teaching staff in the department for their support.

**2015-2019**

**NATIONAL INSTITUTE OF TECHNOLOGY**

**ANDHRA PRADESH**



**CERTIFICATE**

This is to certify that the project report entitled “**ONLINE LEAVE SUBMISSION**”submitted by **K.CHELSEA JYOTSNA** (Rollno:411534), **A.LAVANYA** (Roll no:411503) , **D.RAMYA SREE(** 411520) students of B.Tech CSE to National Institute Of Technology , Andhra Pradesh is a record of bonafied work carried out by her under my guidance. The project fulfils the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission.

**Smt.G.JAYA LAKHSMI HEAD OF DEPARTMENT**

Ad’hoc faculty COMPUTER SCIENCE AND ENGINEERING

National Institute of Technology NATIONAL INSTITUTE OF TECHNOLOGY

ANDHRA PRADESH ANDHRA PRADESH

**Table of Contents**

1. **Introduction**

1.1 Purpose

1.2Project Scope

1.3References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Function Overview

2.3 User Characteristics

**3. Functional Requirements**

3.1 Inputs

3.2 Processing

3.3 Outputs

**4.External Interface Requirements**

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

**5. Software Design**

5.1 Class Diagram

5.2 Deployment Diagram

5.3 Data Flow Diagram

**6. Code Templates**

**7. Testing**

**8. Screenshots of the Project**

**9. Conclusion**

**10. Future Work**

**LIST OF FIGURES**

FIGURE NAME PAGE NO

1. CLASS DIAGRAM 11

2. SEQUENCE DIAGRAM 11

3. DATA FLOW DIAGRAM 12

4. HOME PAGE 37

5. SIGNUP FORM 38

6. LOGIN 38

7. LEAVE FORM 39

8. PROFILE 39

9. MAIL SENT 40

10. PHP DATABASE FOR STUDENTS 40

11. PHP DATABASE FOR ROLLNOS 40

**ABSTRACT**

This project ONLINE LEAVE SUBMISSION is very useful to the students/employees to apply for leave easily to the respective authorities they want to apply for leave.This system presently we are sending the leave forms through mails but we will develop this to be a separate system where there will be no need of internet to submit or to send leaves.RThus we want to make our project as offline system so that the people will do it easily.This project we developed in an educational institute level. Here students or staff may apply leave to the coordinator and the coordinator may accept/reject the leave and the return leave will be sent back to the student.

# SRS FOR ONLINE LEAVE SUBMISSION

# INTRODUCTION:

The objective of the online leave submission is to maintain the details of the user

regarding their departing schedule and to request for the permission from higher authority.

The permission for granting leave is issued by higher authorities through online based

on the user's previous leave history.

## 1.1 PURPOSE:

The purpose of the documents is to provide the functional and non-functional requirements

of the online leave submission which helps the manager to know the functionalities which

are going to be implemented.

## 1.2 SCOPE:

The scope of online leave system is up to that institute level where the higher authority can

grant the permission for leave when the employee or student wants to take leave. It is highly

helpful when the higher authority is not in station at that time.

## 1.3 REFERENCES:

Http://www.youtube.com/playlist?list=PLr6-GrHUIVF\_ZNmuQSXdS1970yr1L9sPB

Http://www.youtube.com/playlist?list=PLr6-GrHUIVf8JIgLcu3sHigvQjTw\_aC9C

Http://www.youtube.com/playlist?list=PLRGaFnSHWrjkpK2zD4TWKWMWVfeYK-b

# 2. OVERALL DESCRIPTION:

## 2.1 PRODUCT PERSPECTIVE:

The user can be able to know still how many leaves he can apply in online through this website. And the higher authority may grant or reject permission based on the issue for which the user mentioned his/her reason for leave.

## 2.2 PRODUCT FUNCTIONS OVERVIEW:

The users who belong to that institute have to register in this website and have to create

an account by filling his/her details. Whenever the user wants permission from the higher

authority he/she should login into his account and submit the leave form which have details

such as to date, from date, reason for permission, to the higher authority from whom he

wants to take permission. If the higher authority accepts or rejects the permission that

notification message should reach the user.

## 2.3 USER CHARACTERISTICS:

The system includes three modules which are as follows:

### 2.3.1 ADMIN MODULE:

1. Admin should always check whether the website is getting misused anywhere.

2. He also checks and corrects the inconsistency of the database.

## 2.3.2 USER MODULE:

1. Creates an account and login into account and applies for permission whenever needed.

### 2.3.3 HIGHER AUTHORITY MODULE:

1. Checks notifications from this website and based on the reason of the user's issue

he accepts/rejects permission.

# 3. FUNCTIONAL REQUIREMENTS:

## 3.1 INPUTS:

1. Students database consisting of his details, leave history and his to date (from which

date he wants to take), from date(till which date), reason for permission as inputs.

## 3.2 PROCESSING:

1. Data mining is to be done on the student's database and check if the number of leaves

taken by the user has exceeded the maximum value of leaves or not.

2.If the number is exceeded then an error message is sent to the user and he cant apply

for permission through website.

## 3.3 OUTPUTS:

1. If the higher authority grants/rejects that permission then the notification message is

sent to the user as permission granted/rejected.

2.If the number of leaves exceeded the maximum level, then an error message is sent to

user.

# 4. EXTERNAL INTERFACE REQUIREMENTS:

## 4.1 USER INTERFACE:

The user interface will be implemented by using HTML, CSS, JAVASCRIPT, PHP, MYSQL.

## 4.2 HARDWARE INTERFACE:

Databases for storage where the database is composed of students details, their leaves

history.

## 4.3 SOFTWARE INTERFACE:

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets

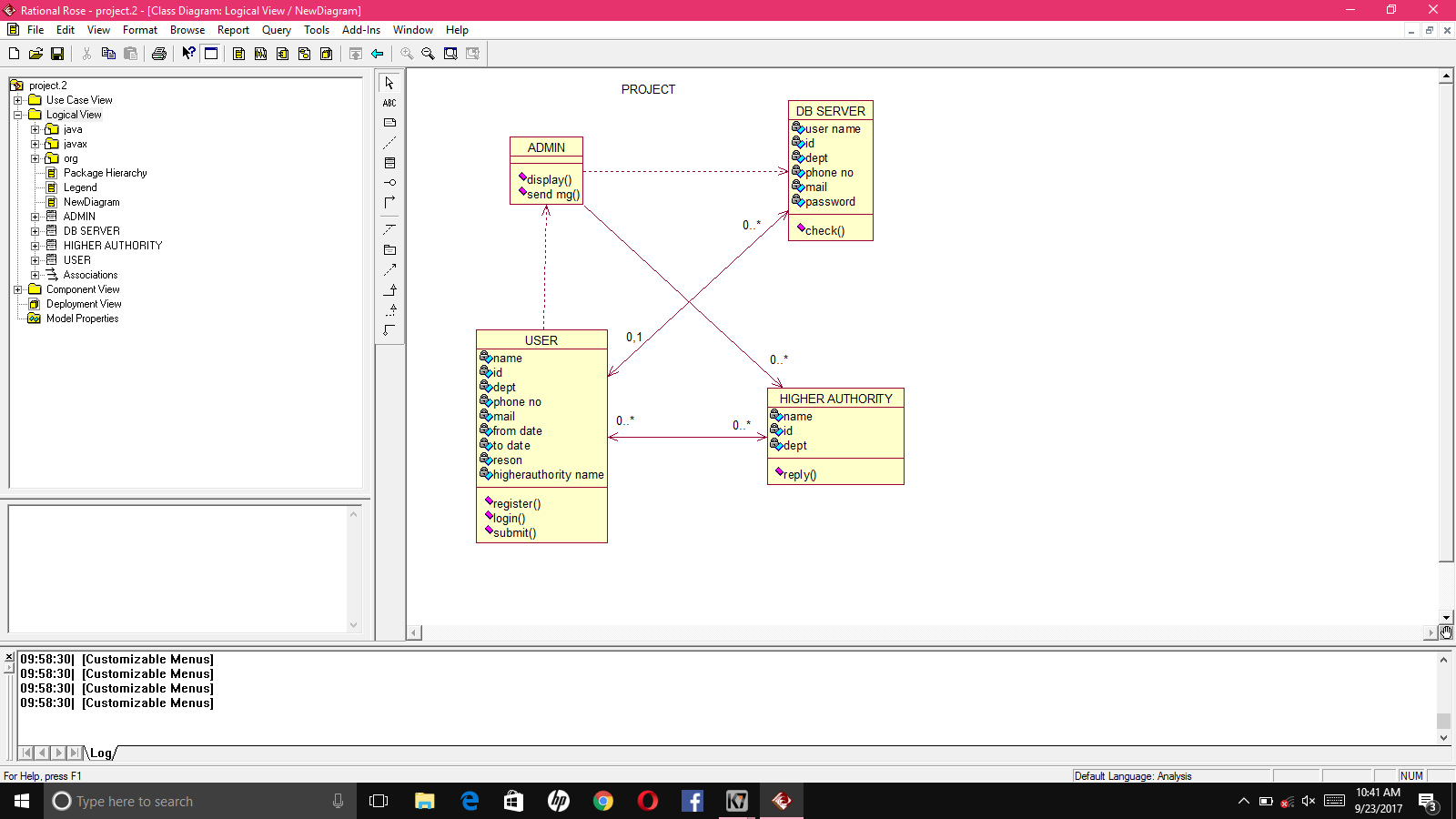
JAVASCRIPT

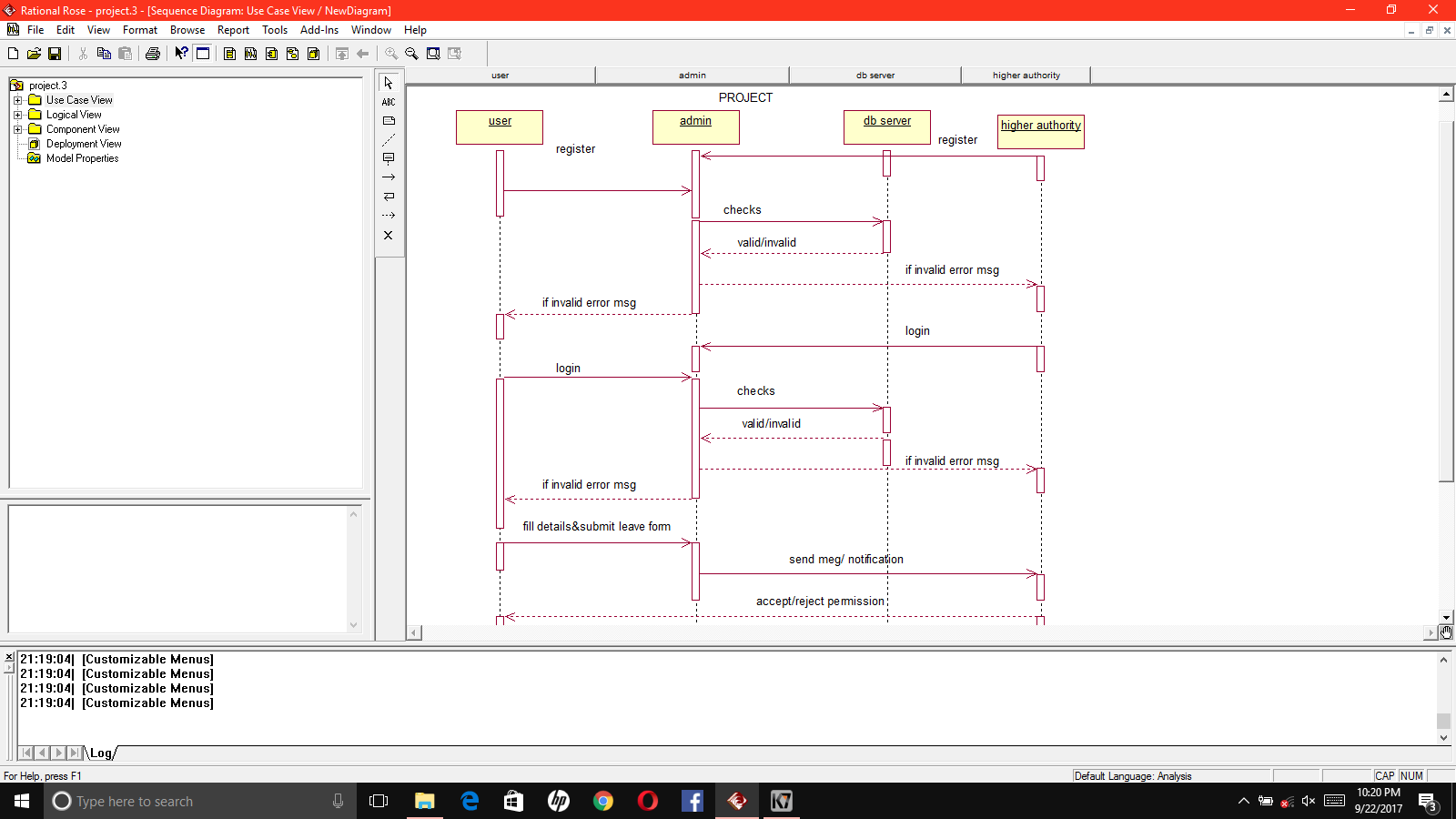
PHP: HyperText Preprocessor

MYSQL: Structured Query Language

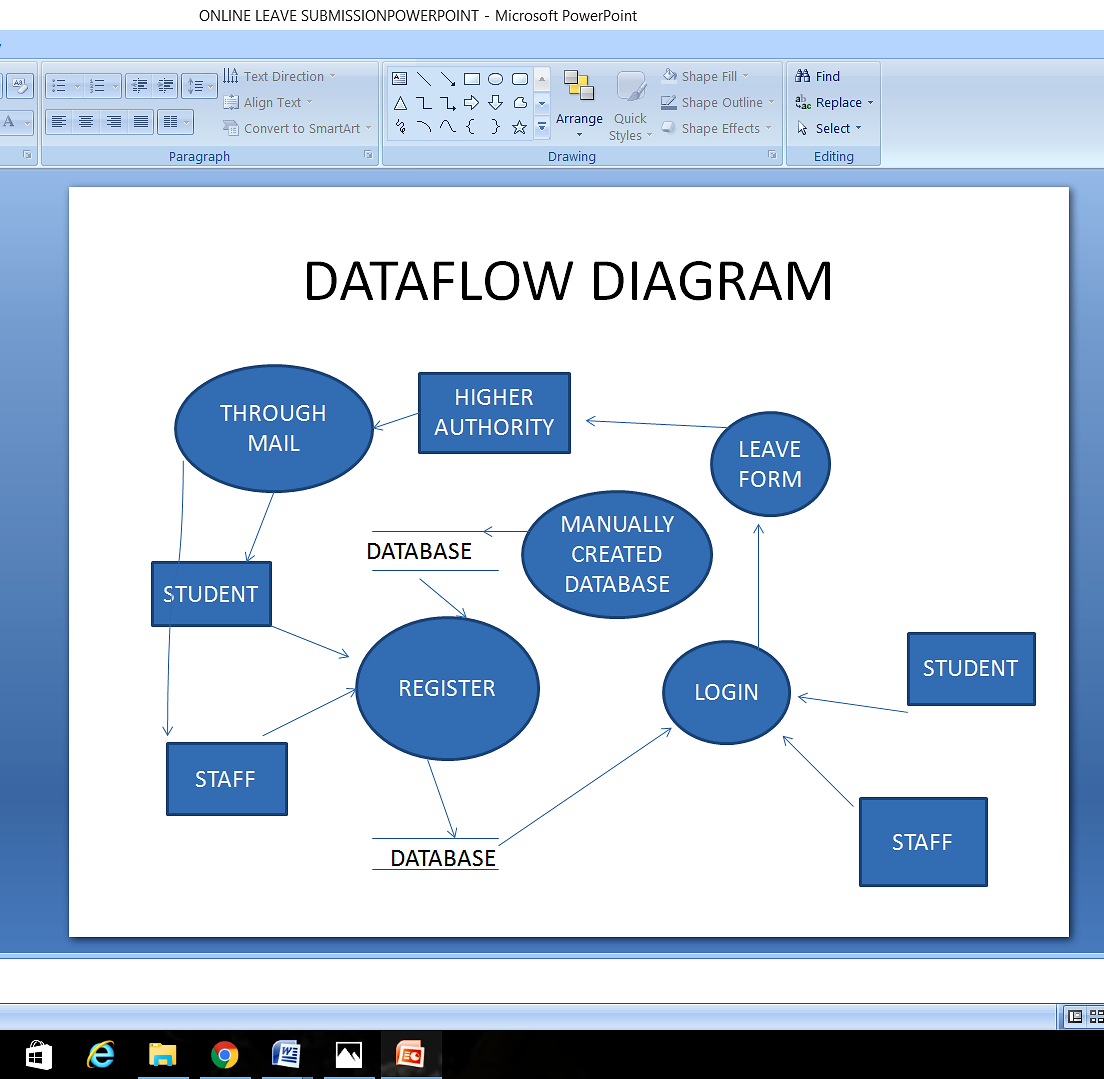
## 5.SOFTWARE DESIGN:

## 5.1CLASS DIAGRAM:



**5.2 SEQUENCE DIAGRAM**:****

## 5.3 DATA FLOW DIAGRAM:



6.CODE TEMPLATES

6.1.CODE FOR MAIN WEBSITE

<!doctype html>

<html>

<head>

<link type="text/css" rel="stylesheet" href="stylesheet.css">

</head>

<body>

<div style="width:100%;height:150px;">

<img id="home" style="float:left;width:12%;height:140px,margin-top:0px;" src="NIT-Andhra-Logo.ico"></img>

<h1 style="font-family:'elephant italic';color:white;font-style:verdana;margin-top:0px;font-size:60px;padding:10px 0 0 20px;text-align:center;

position:relative;top:50%;-webkit-transform:translateY(-50%);-ms-transform:translateY(-50%);-transform:translateY(-50%);">ONLINE&nbsp LEAVE&nbsp SUBMISSION</h1></div>

<ul>

<ul id="ul1">

<li class="dropdown">

<a href="javascript:void(0)" class="dropbtn">LOGIN</a>

<div class="dropdown-content">

<a href="login1.html">student</a>

<a href="login2.html">staff</a>

<a href="login3.html">administration</a>

</div>

</li>

<li class="dropdown">

<a href="javascript:void(0)" class="CONTACT">SIGN UP</a>

<div class="dropdown-content">

<a href="signup1.html" target="\_blank">student</a>

<a href="signup2.html" target="\_blank">staff</a>

<a href="signup3.html" target="\_blank">administration</a>

</div>

</li>

<li class="dropdown">

<a href="javascript:void(0)" class="CONTACT">CONTACT</a>

<div class="dropdown-content">

<a href="nitapoffice.html" target="\_blank">NIT office</a>

</div>

</li>

<li class="dropdown">

<a href="javascript:void(0)" class="CONTACT">ABOUT US</a>

<div class="dropdown-content">

<a href="contactusadmin.html" target="\_blank">administrator</a>

</div>

</li>

</ul>

<div class="slideshow-container">

<div class="mySlides fade">

<div class="numbertext">1 / 2</div>

<img src="file://C:/xampp/htdocs/jyo/newclass.jpg" style="width:100%;height:500px">

<div class="text">NIT ANDHRA MAIN BUILDING1</div>

</div>

<div class="mySlides fade">

<div class="numbertext">2 / 2</div>

<img src="file://C:/xampp/htdocs/jyo/nitapmain.jpg" style="width:100%;height:500px">

<div class="text">NIT ANDHRA BUILDING2</div>

</div>

</div>

<br>

<div style="text-align:center">

<span class="dot"></span>

<span class="dot"></span>

</div>

<script>

var slideIndex = 0;

showSlides();

function showSlides() {

var i;

var slides = document.getElementsByClassName("mySlides");

var dots = document.getElementsByClassName("dot");

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none";

}

slideIndex++;

if (slideIndex> slides.length) {slideIndex = 1}

for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "");

}

slides[slideIndex-1].style.display = "block";

dots[slideIndex-1].className += " active";

setTimeout(showSlides, 3500); // Change image every 2 seconds

}

</script>

<div class="main">

<p>This is all about online leave submission.We are here to send your leaves in a formatted order to higher officials.</p>

<p>This helps us in many ways.This is institutuional level</p>

</div>

<div id="footer">

<span style="color:black;font-size:90%;font-family:'verdana';padding:10px;text-align:center;background-color:white;display:block;margin-left:auto;margin-right:auto;">&copy 2017 NITAP.COM All Rights Reserved</span>

</div>

</body>

<html>

STYLE SHEET FOR MAIN WEBSITE

body{

background-color:gray;

margin:0px;

font-family:verdana,Georgia;

}

#header{

background-color: #F1F1F1;

text-align: center;

padding: 10px;

font-size:50px;

}

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333;

clear:both;

}

li {

font-size:20px;

float: left;

}

li a, .dropbtn {

display: inline-block;

color: white;

text-align: center;

padding: 14px 50px;

text-decoration: none;

}

li a:hover, .dropdown:hover .dropbtn {

background-color:rgb(0,133,68);

}

li.dropdown {

display: inline-block;

}

.dropdown-content {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

z-index: 1;

}

.dropdown-content a {

color: black;

padding: 12px 20px;

text-decoration: none;

display: block;

text-align: left;

}

.dropdown-content a:hover {background-color:rgb(217, 165, 243)}

.dropdown:hover .dropdown-content {

display: block;

}

.aboutus a:link,.aboutus a:visited {

background-color: black;

color: white;

padding-right:100px;

text-decoration: none;

display: inline-block;

}

.aboutus a:hover,.aboutus a:active {

background-color: black;

color:white;

}

\* {box-sizing:border-box}

body {font-family: Verdana,sans-serif;}

.mySlides {display:none}

/\* Slideshow container \*/

.slideshow-container {

max-width: 1000px;

position: relative;

margin: auto;

}

/\* Caption text \*/

.text {

color: #f2f2f2;

font-size: 15px;

padding: 8px 12px;

position: absolute;

bottom: 8px;

width: 100%;

text-align: center;

}

/\* Number text (1/3 etc) \*/

.numbertext {

color: #f2f2f2;

font-size: 12px;

padding: 8px 12px;

position: absolute;

top: 0;

}

/\* The dots/bullets/indicators \*/

.dot {

height: 13px;

width: 13px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active {

background-color: #717171;

}

/\* Fading animation \*/

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 1.5s;

animation-name: fade;

animation-duration: 1.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

/\* On smaller screens, decrease text size \*/

@media only screen and (max-width:500px) {

.text {font-size: 11px}

img{

max-width:50%;

max-height:50%;

display:inline;

margin-left:0px;

margin-right:0px;

float:left;

}

.left{

float:left;

}

.right{

float:right;

}

.signup a:link,.signup a:visited {

background-color: black;

color: white;

padding: 14px 25px;

text-align: center;

text-decoration: none;

display: inline-block;

}

.signup a:hover,.signup a:active {

background-color: red;

color:white;}

6.2. CODE FOR SIGNUP

<!doctype html>

<html>

<head>

<title>signup form</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<h1>Signup Form</h1>

<form name="registration" method="post" action="register1.php" >

<input type="text" placeholder="Firstname" name="fname" pattern="[A-Za-z]{6,}" title="minimum six letter" required>

<input type="text" placeholder="Lastname" name="lname"pattern="[A-Za-z]{6,}" title="mminimum six letter" required>

<input type="email" placeholder="Email" name="email" required>

<input type="text" placeholder="username" name="uname"pattern="[A-Za-z0-9]{6,}" title="minimum length 6(characters or numbers)" required>

<input type="password" placeholder="Password" pattern="[A-Za-z0-9]{6,}" name="pass" title="minimum length 6(characters or numbers)" required>

<input type="submit" name="Signup">

</form>

</body>

</html>

6.3.CODE FOR LOGIN

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Login Page</title>

<style>

/\* Basics \*/

html, body

{

padding: 0;

margin: 0;

width: 100%;

height: 100%;

font-family: "Helvetica Neue" , Helvetica, sans-serif;

background: #FFFFFF;

}

.logincontent

{

position: fixed;

width: 400px;

height: 350px;

top: 50%;

left: 50%;

margin-top: -150px;

margin-left: -175px;

background: #07A8C3;

padding-top: 10px;

}

.loginheading

{

border-bottom: solid 1px #ECF2F5;

padding-left: 18px;

padding-bottom: 10px;

color: #ffffff;

font-size: 20px;

font-weight: bold;

font-family: sans-serif;

text-align:center;

}

label

{

color: #ffffff;

display: inline-block;

margin-left: 18px;

padding-top: 10px;

font-size: 15px;

}

input[type=text], input[type=password]

{

font-size: 14px;

padding-left: 10px;

margin: 10px;

margin-top: 12px;

margin-left: 18px;

width: 300px;

height: 35px;

border: 1px solid #ccc;

border-radius: 2px;

box-shadow: inset 0 1.5px 3px rgba(190, 190, 190, .4), 0 0 0 5px #f5f5f5;

font-size: 14px;

}

.loginbtn

{

float: right;

margin-right: 20px;

margin-top: 20px;

padding: 6px 20px;

font-size: 14px;

font-weight: bold;

color: #fff;

background-color: #07A8C3;

background-image: -webkit-gradient(linear, left top, left bottom, from(#07A8C3),

to(#6EE4E8));

background-image: -moz-linear-gradient(top left 90deg, #07A8C3 0%, #6EE4E8 100%);

background-image: linear-gradient(top left 90deg, #07A8C3 0%, #6EE4E8 100%);

border-radius: 30px;

border: 1px solid #07A8C3;

cursor: pointer;

}

.loginbtn:hover

{

background-image: -webkit-gradient(linear, left top, left bottom, from(#b6e2ff),

to(#6ec2e8));

background-image: -moz-linear-gradient(top left 90deg, #b6e2ff 0%, #6ec2e8 100%);

background-image: linear-gradient(top left 90deg, #b6e2ff 0%, #6ec2e8 100%);

}

</style>

</head>

<body>

<form name="login" method="post" action="loginactual.php" >

<div class="logincontent">

<div class="loginheading">

Login

</div>

<label for="txtUserName">

Username:</label>

<input type="text" id="txtUserName" name="txtUserName" />

<label for="txtPassword">

Password:</label>

<input type="password" id="txtPassword" name="txtPassword" />

<input type="submit" class="loginbtn" value="Login" id="btnSubmit" />

<input type="reset" class="loginbtn" value="Reset"

id="btnsubmit"/></br></br></br></br>

<a href="#" class="show\_hide">forgotpassword?</a>

</div>

</div>

</form>

</body>

</html>

6.4. PHP FILE FOR SIGNUP

<?php

$fname = filter\_input(INPUT\_POST, 'fname');

$lname = filter\_input(INPUT\_POST, 'lname');

$email = filter\_input(INPUT\_POST, 'email');

$uname = filter\_input(INPUT\_POST, 'uname');

$pass= filter\_input(INPUT\_POST, 'pass');

if(!empty($uname)){

if(!empty($pass)){

$host="localhost";

$dbusername="root";

$dbpassword="";

$dbname="data";

$conn= new mysqli ($host,$dbusername,$dbpassword,$dbname);

if(mysqli\_connect\_error()){

die('Connect Error (' .mysqli\_connect\_error() .')'

. mysqli\_connect\_error());

}

else{

$sql="select \* from student\_id where ROLLNO='$uname' ";

if($result=mysqli\_query($conn,$sql))

{

if(mysqli\_num\_rows($result)>0)

{

$sql="select user\_name from student where user\_name='$uname'";

if($result=mysqli\_query($conn,$sql))

{

if(mysqli\_num\_rows($result)>0){

echo "username already exist";

}

else{

$sql="INSERT INTO student (First\_name,Last\_name,Email\_id,user\_name,password) values('$fname','$lname','$email','$uname','$pass')";

if ($conn->query($sql)){

echo "New record is inserted successfully”

}

else{

echo "Error :" . $sql ."<br>" . $conn->error;

}

$conn->close();

}

}

}

else{

echo "rollno does not exist";

}

}

else{

echo "Error :" . $sql ."<br>" . $conn->error;

}

// $conn->close();

}

}

else{

echo "password should not be empty";

die();

}

}

else{

echo "username should not be empty";

}

?>

6.5. PHP CODE FOR LOGIN

<?php

$uname=filter\_input(INPUT\_POST,'txtUserName');

$pass=filter\_input(INPUT\_POST,'txtPassword');

if(!empty($uname))

{

if(!empty($pass))

{

$host="localhost";

$dbusername="root";

$dbpassword="";

$dbname="data";

$conn= new mysqli ($host,$dbusername,$dbpassword,$dbname);

if(mysqli\_connect\_error())

{

die('Connect Error (' . mysqli\_connect\_error() .')' . mysqli\_connect\_error());

}

else

{

$sql="select \* from student where user\_name='$uname' and password='$pass' ";

if($result=mysqli\_query($conn,$sql))

{

if(mysqli\_num\_rows($result)>0)

{

echo "successfully logged in!";

echo <<<HTML

<a href="leave123.html">APPLY LEAVE</a>

HTML;

}

else

{

echo "user does not exist";

}

}

else

{

echo "Error:" . $sql . "<br>" . $conn->error;

}

$conn->close();

}

}

else

{

echo "password should not be empty";

}

}

else

{

echo "username should not be empty";

die();

}

?>

6.6. CODE FOR LEAVE FORM

<!doctype html>

<html>

<head>

<title>leave form</title>

<link rel="stylesheet" type="text/css" href="style1.css"/>

</head>

<body>

<h1>Leave Form</h1>

<form method="POST" action="mailpro1.php">

<label for="username">Name:</label>

<input type="text" id="username" name="username" required>

<label for="idno">Rollno:</label>

<input type="text" id="idno" name="idno"required>

<label for="email">Email\_id:</label>

<input type="email" id="email" name="email"required>

<label for="year">Year of Studying:</label>

<select >

<option value="select year" >select year</option>

<option value="I-year">I-year</option>

<option value="I-year">II-year</option>

<option value="I-year">III-year</option>

<option value="I-year">IV-year</option>

</select>

</br></br>

<label for="branch" >Branch:</label>

<select >

<option value="select dept">select dept</option>

<option value="cse">CSE</option>

<option value="I-year">ECE</option>

<option value="I-year">EEE</option>

<option value="I-year">MECH</option>

<option value="select-year">CIVIL</option>

<option value="I-year">BIO-TECH</option>

<option value="I-year">CHEMICAL</option>

<option value="I-year">METALLURGY</option>

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

<label for="HNAME">HigherAuthority Name:</label>

<input type="text" id="HNAME" name="HNAME"required>

<label for="reason">Reason:</label>

<input type="text" id="reason" name="reason" required>

FromDate:

<input type="date" name="fromday" required></br></br>

ToDate:

<input type="date" name="today" required></br></br>

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

</form>

</body>

</html>

7.TESTING

softwate testing:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended environments, and
* achieves the general result its stakeholders desire.

white box testing:

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs.

White-box test design techniques include the following code coverage criteria:

Control flow testing

Data flow testing

Branch testing

Statement coverage

Decision coverage

Modified condition/decision coverage

Prime path testing

Path testing

blackbox testing:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

Test procedures:

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required..[2] The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.[3]

Test cases:

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

Test design techniques:

Typical black-box test design techniques include:

Decision table testing

All-pairs testing

Equivalence partitioning

Boundary value analysis

Cause–effect graph

Error guessing

State transition testing

Use case testing

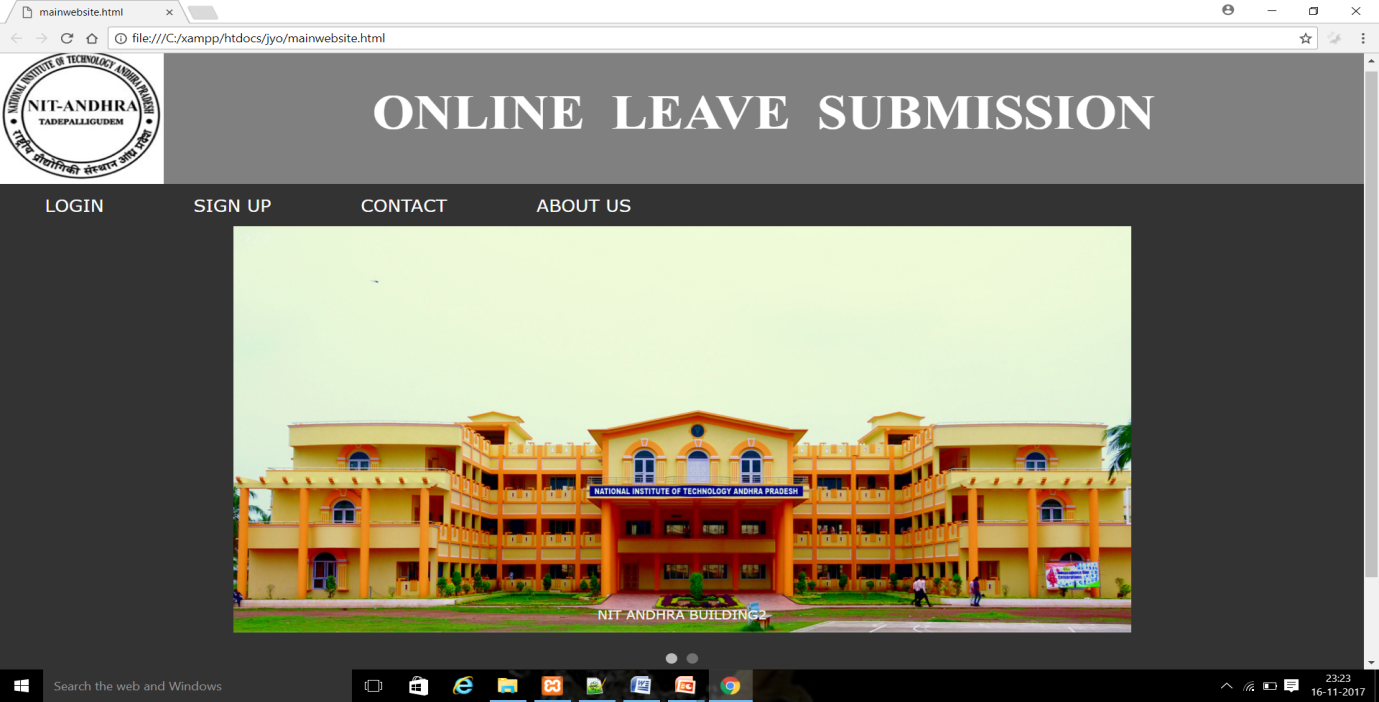
User story testing

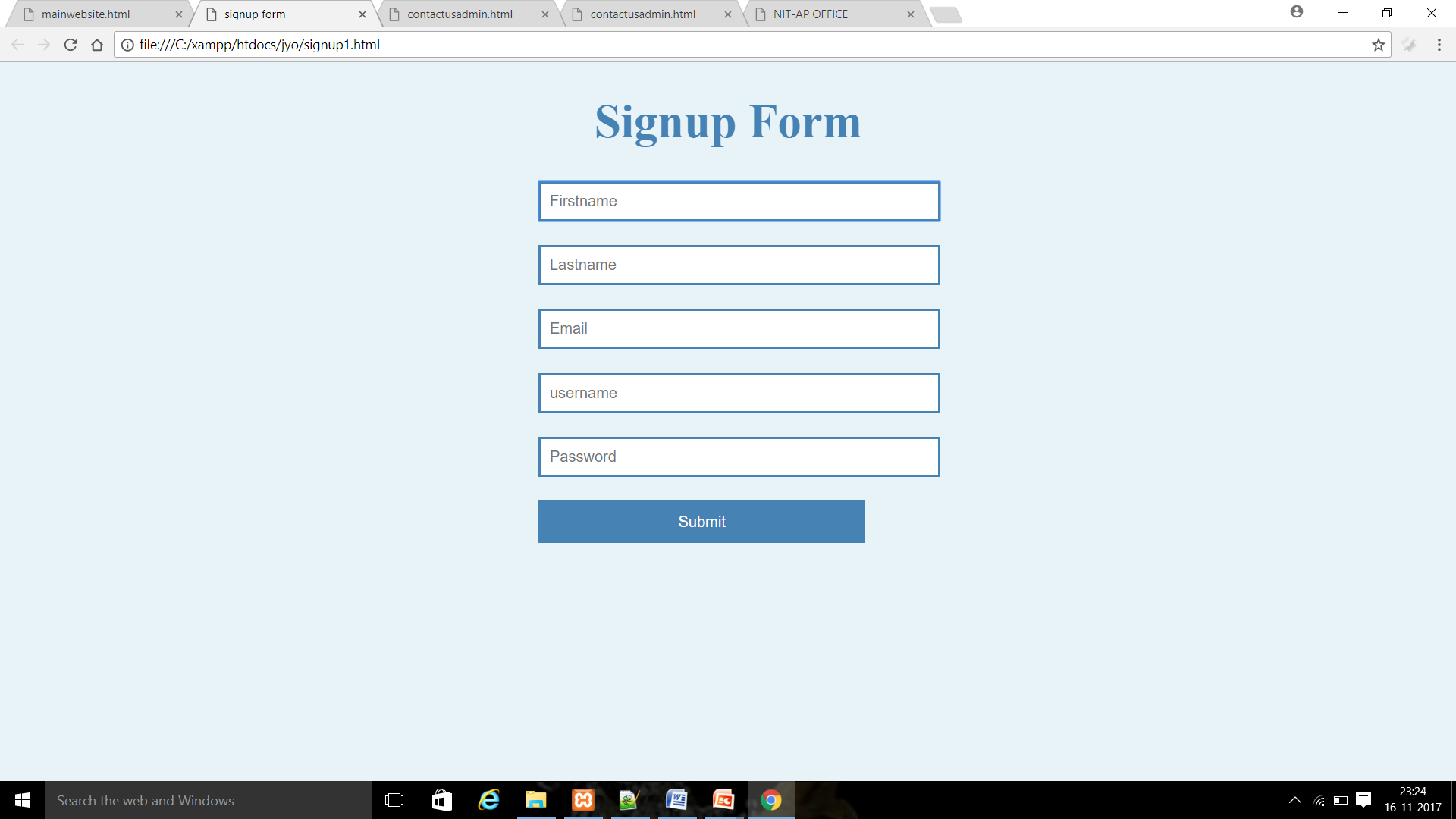
Domain analysis

Syntax testing

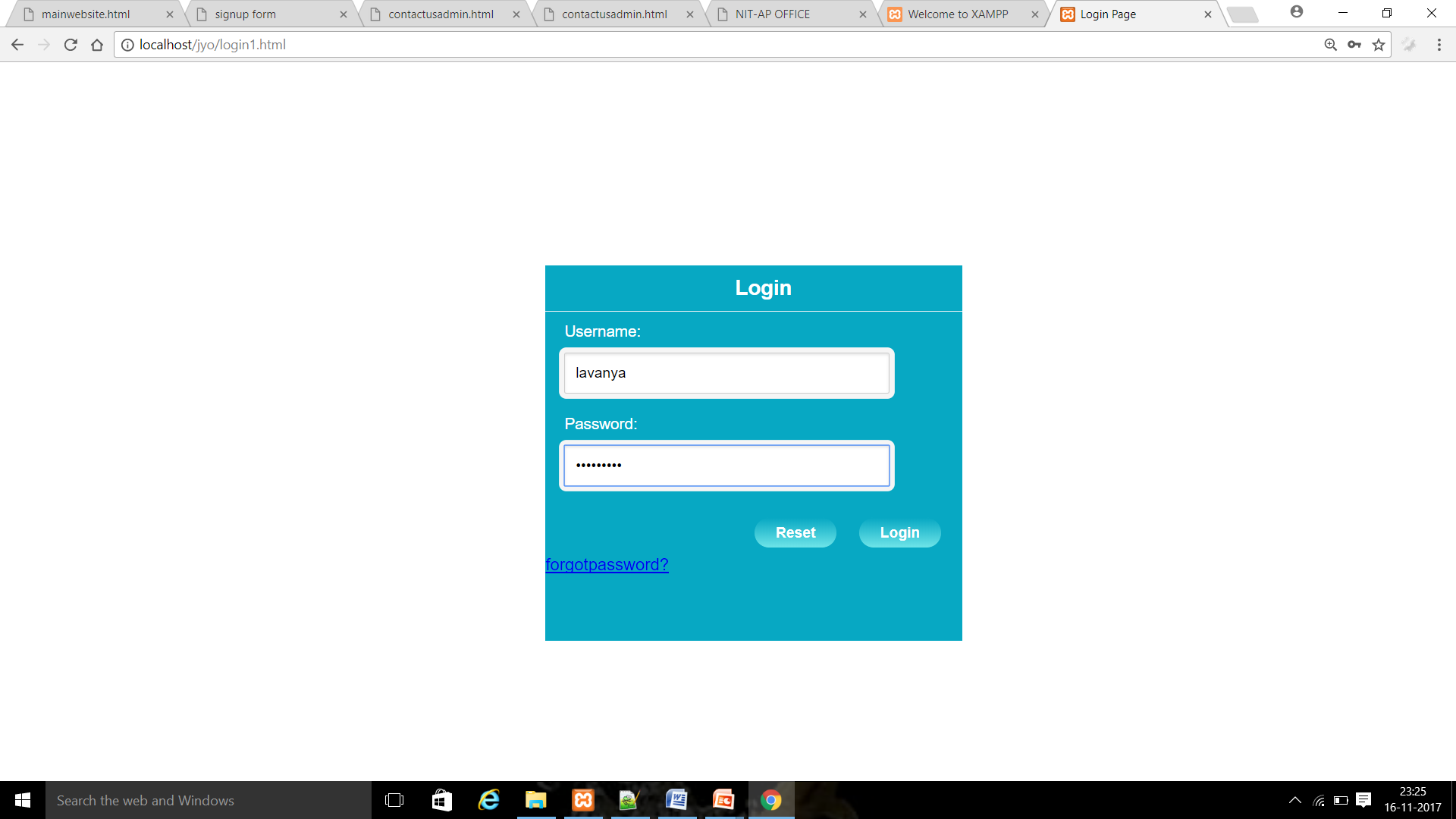
Combining technique

**8.SCREENSHOTS OF OUR PROJECT:**

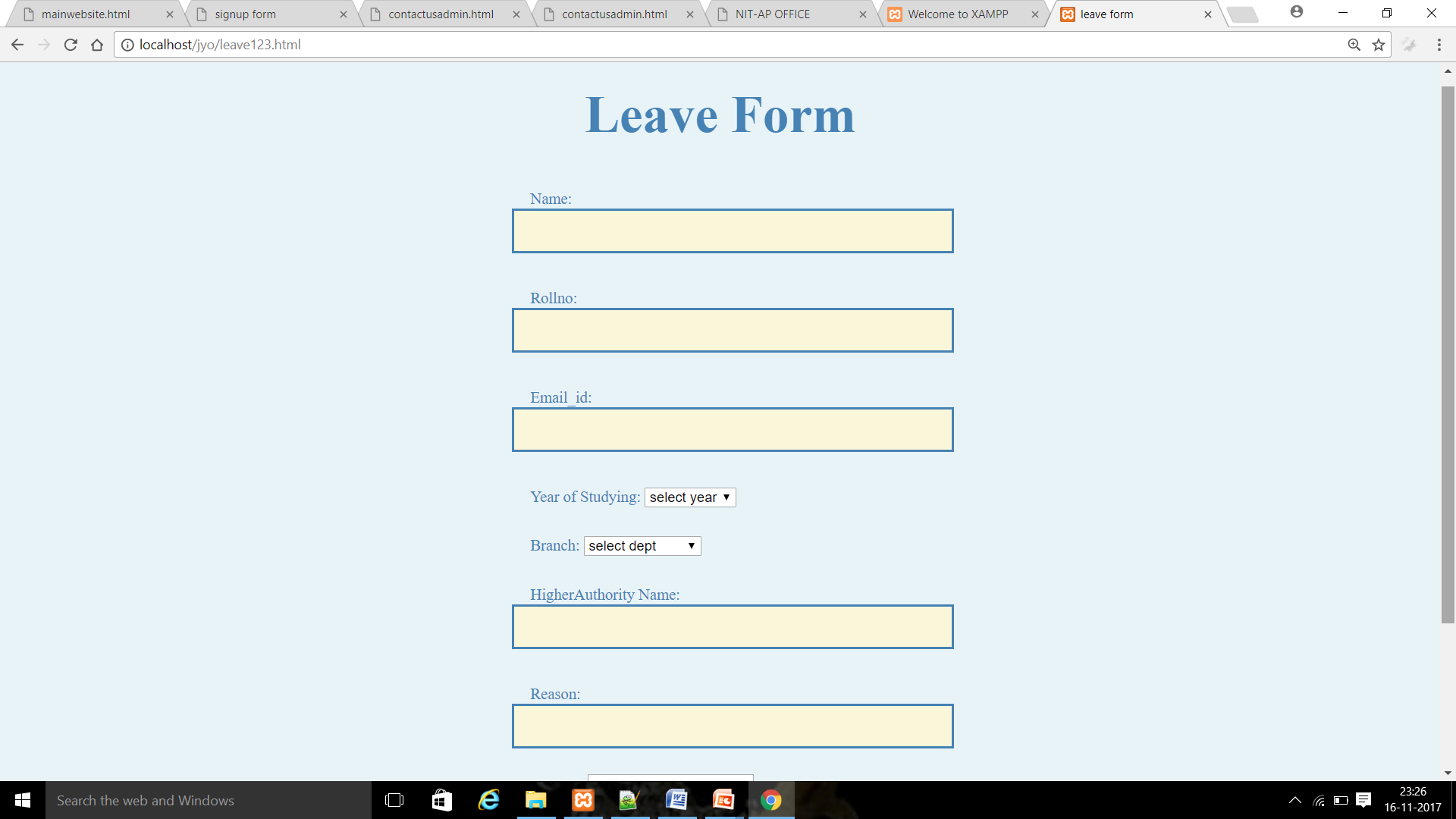
****

**SIGNUP FORM:**

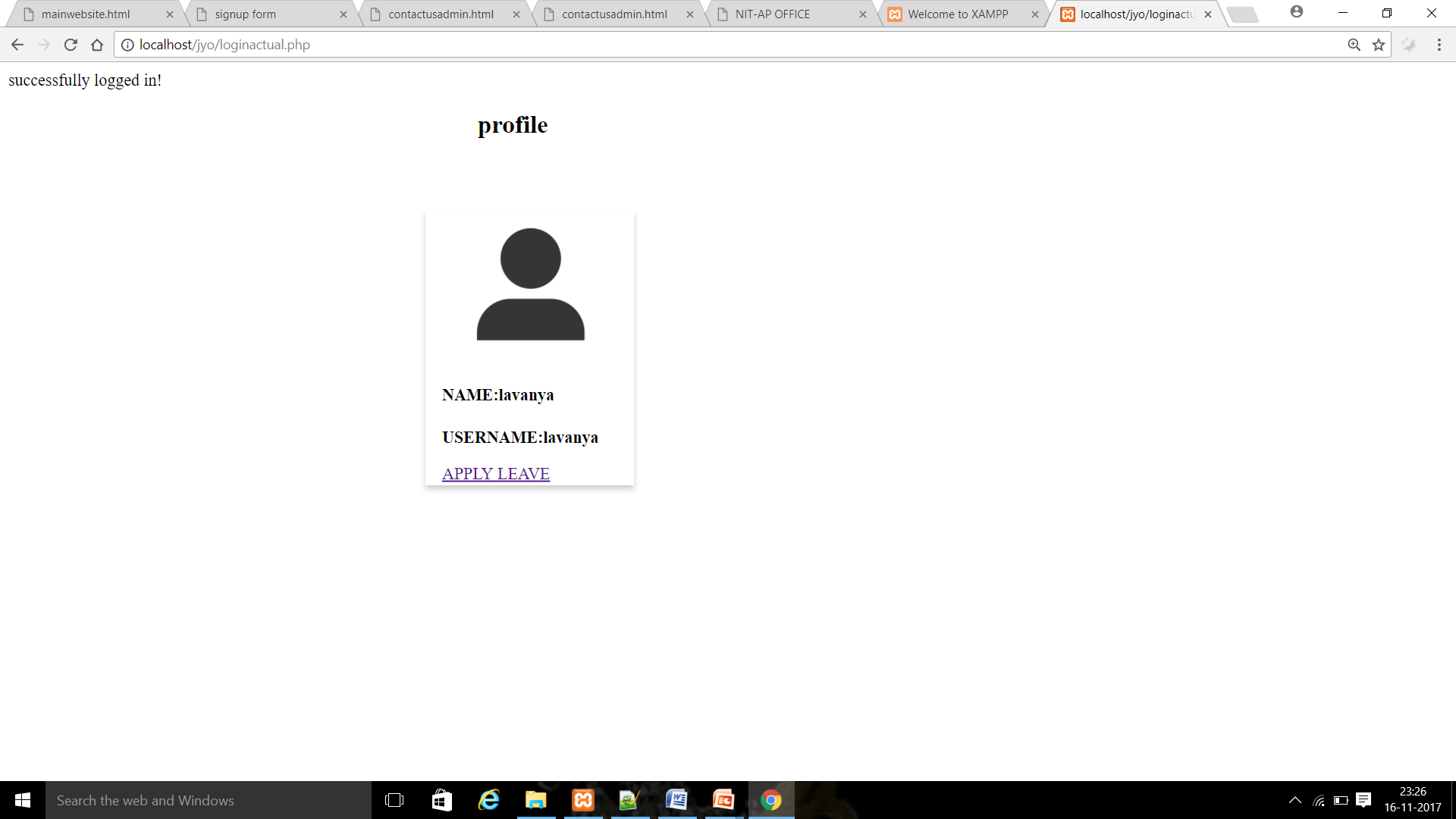
**LOGIN PAGE:**

****

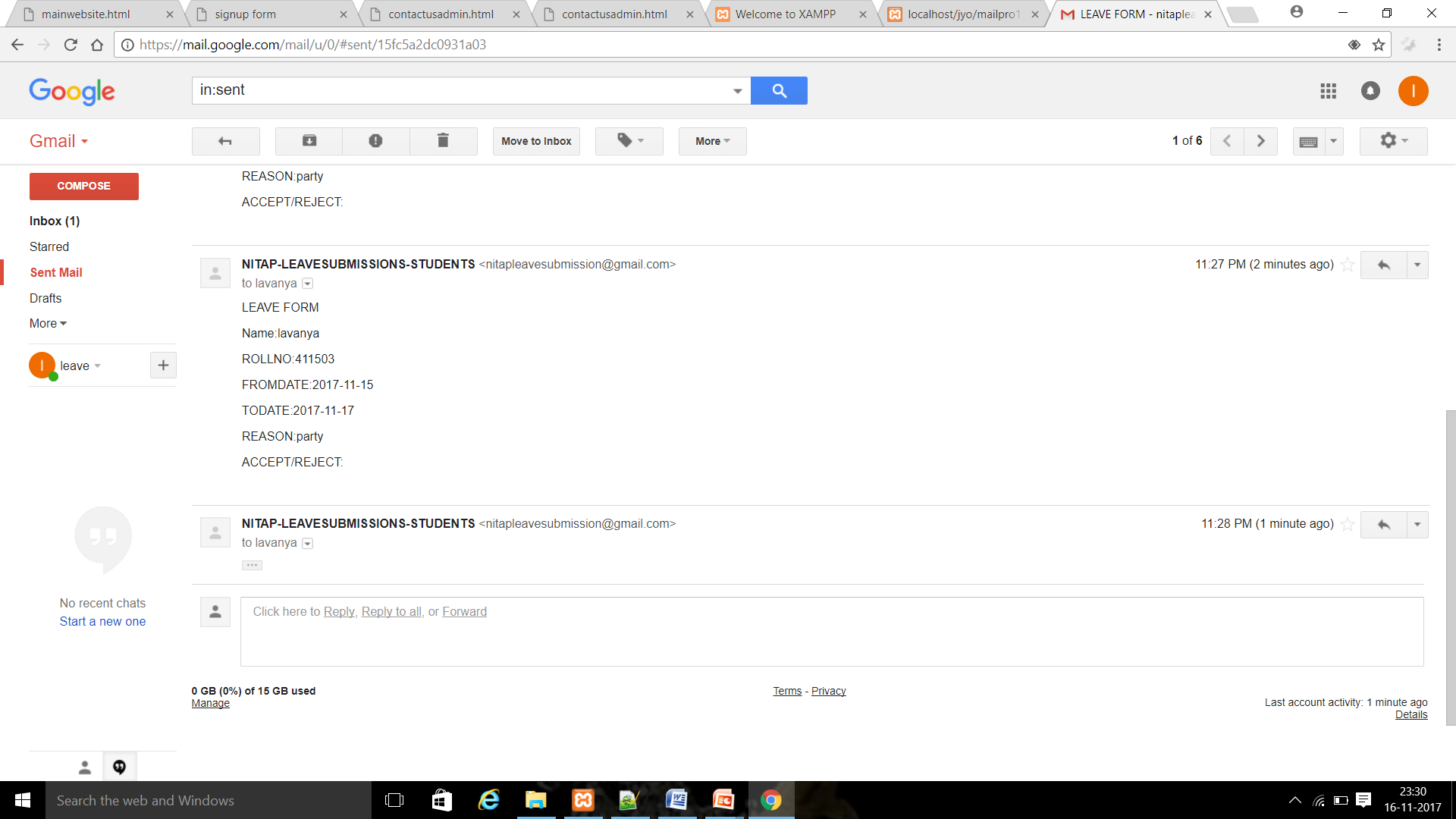
## LEAVE FORM:



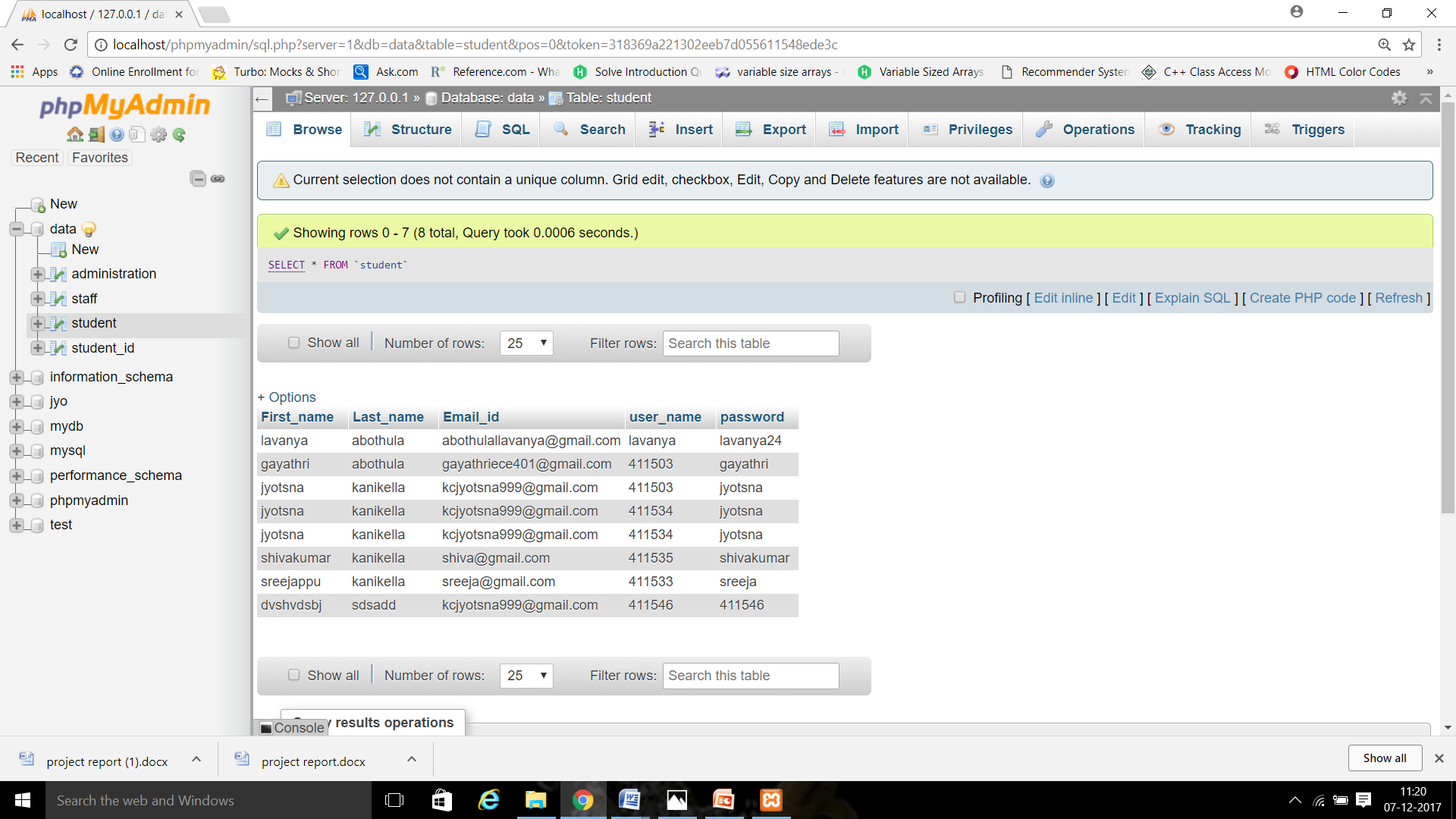
**PROFILE PAGE:**

****

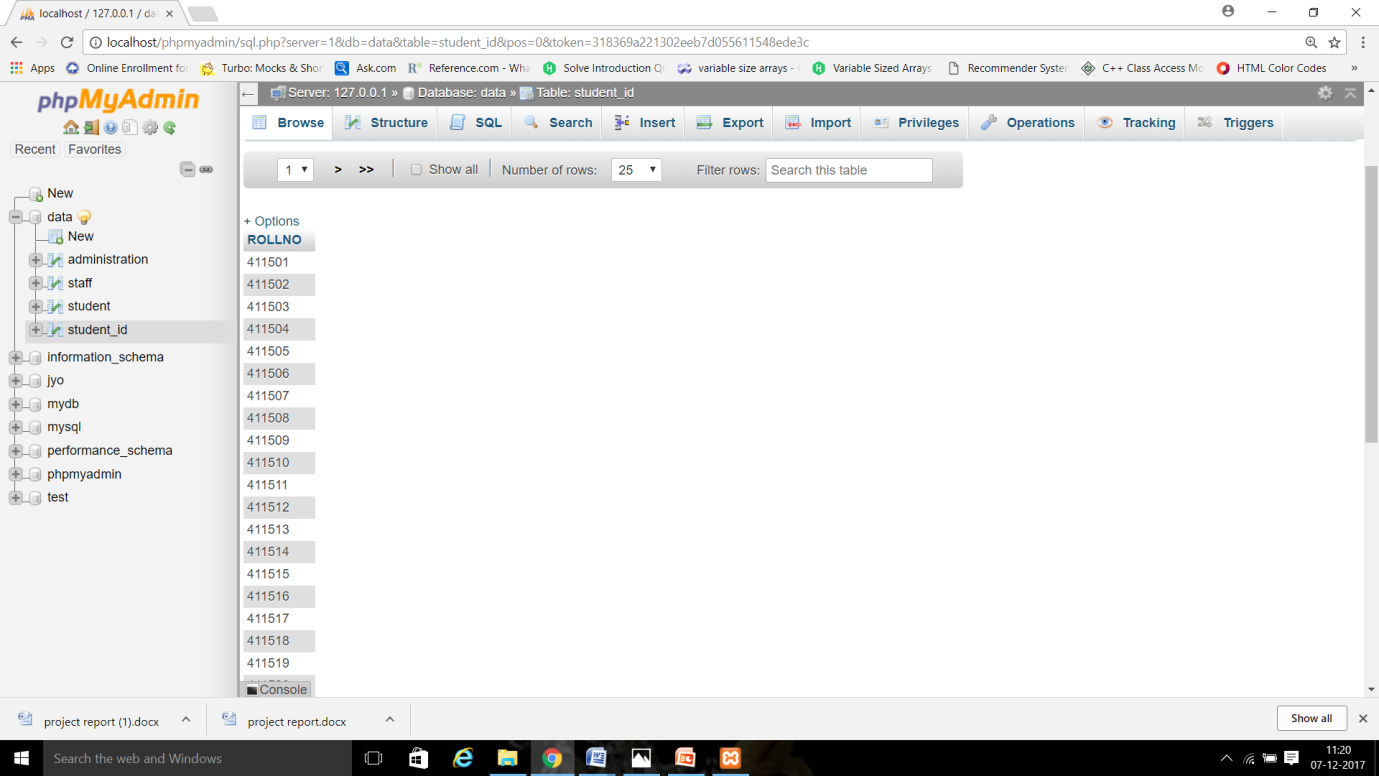
## MAIL SENT:



**PHP DATABASE FOR STUDENTS:**

****

## PHP DATABASE FOR ROLL NO S:



9. CONCLUSION

The project is so helpful for students, staff, coordinator. Whenever coordinator is in out of station this project is definitely of great help. As an engineer we tried to help the students and the administration and students or administration and employees in industrial and institutional level to eliminate their discomforts while applying leave.

10. FUTURE WORK

As a part of future work we are concentrating on certain aspects like the respective professor may send the leave to his student, he can send the leave to the respective student as a part of our project. And another important thing is that if the student already took more than limited leaves, in the profile page when he clicks on apply leave link it should show oops!! Cant apply.

Also the profile page should show students picture also it show details like how may leaves they applied, how many accepted, how many rejected also.

The administration can also check the individual accounts so that they will know about the leave history of the students. We are also trying to make this project into a separate domain where messages are sent automatically without the usage of the internet