**Topic :-> PHP Simple To-Do List**

* ***ABSTRACT*** :-

This is a Web Development application where a user can add a task in the application which he/she need’s that task to be done, he/she can mark the task as done when they have completed the work. Now the task may be as Simple as bringing Groceries to Many Important task’s Such as Bringing Medicine. There may Sometime that you do not want other’s to look at the task’s added except for yourself so we did add a login page where once you logout no one can login except you which can ensure that your task’s aren’t leaked. This application is built on three main things that are HTML This is a Markup language that means that this language is used as the GUI which is being interacted by the user. The next main thing is the PHP it is a server-side Scripting Language where this language would take your input from the user it will then used for storing in the database which is the third part where we can store the data needed and delete the data which when done. This application provides a smooth and efficient way of manipulation of the task’s and to top off this application is also scalable where it can be used for managing list of larger size.

* ***OBJECTIVE***:-

The main reason as to why I built this application was to help the user’s stay organized and on top of their tasks. This application can be used mainly for “Increasing productivity” which would help users to stay focused on their tasks and avoid distractions. By Having a clear list of what needs to be done ,users can be better prioritize their time and energy. Then we have another reason which is “Reduced stress and anxiety” now when you have a workload amount of tasks assigned you need to have a diary which can help you keep track as to where you are with your task. Now in olden days people had diary’s to keep the track of task but in this modern world people need more reliable and efficient way of handling the task. We also took in consideration of the users “Time Management” why? Because when you have a list of tasks at your hand the very first thing you do is prioritize as to which task has to be done first and which task is to be taken care after. By having a clear plan for what needs to be done the users can avoid wasting time on unnecessary tasks or activities. Another main reason behind building this project is that the user can increase the sense of accomplishment where this application can help users feel more accomplished by giving them a visual representation of their list, they can see how much they have achieved. The Overall reason behind building this application is to make users more productive reduce the stress and anxiety, improve their time management skills and feel more accomplished. These are the main things that we kept in mind and we actually built the application on that we also included a login page where the user will login in to the application then he will be asked to go to To-do List and then he will be redirected to that page.

* ***INTRODUCTION***:-

Listing application is actually a very simple application whose main motive is to add or assign a list of to do things that need to be done the tasks may be anything like being personal task such as running an errand or cleaning the house ,or for work tasks, such as completing projects or attending meetings, the tasks can be simple or complex ,depending on the individual’s needs and preference .In this era the work load is actually increasing day by day and they need a application where they can add all the tasks now in the olden days people had to write everything down and when something were to happen like their diary in which they had written the tasks were to be gone then they would have no idea as to what they have tasks which can be important. Now a days all the things are getting digitalized where everything is moving online and people are all using the online applications now this website is for people like them who are coming on online. We actually divided this project into 3 parts where the first part would be the “Front-End” or some people also like to say that this is the “GUI ” where the user will interact with the application and will actually add or mark or delete the task according to his wish. The second part is actually the middle ware part where all the things get to work the language which was actually used is “PHP” now the php is actually a server side scripting language now what does that mean that means this actually runs on the server side or you can say the company side will handle all the task after interacting with the user. Then comes the last part “Storage” where all your data is actually stored all the tasks which you are adding and all the tasks that the user is deleting are deleted from the database.

* ***METHODOLOGY :-***

Login Page:-

This is the very first thing that you will see when you fire up the application here you will actually enter your username and your respected password which you assigned to that particular email. This application contains a unique email identity that means suppose you give a email saying [hello123@gmail.com](mailto:hello123@gmail.com) now the other person who will create another email account is not allowed to create the email with the same email name cause it is already taken and you need to create a unique email which is different than the other and then comes the register link where you will have 3 fields the very first field will contain your respected email id through which you will login then your second field will actually contain the password which you will assign for this email in this application that means you do not have to give the same password which you have given to the applications which have your personal details .Once this is done then we will move on to the third field which would be re enter your confirm email password which you entered into the 2nd field and then you will press enter that will send data to the backend where the email which you have provided will be checked with the data contains email if your email matches with the one which is present then you have to change your email id or else if you have a password for that email then you simply need to put that password after your email and you have to hit login button

List Application:-

Now you will be actually be redirected to a page where you will be seen with a Interface where you will have a text field where you will have to give the task which you have to do and then you will find a add item button just down of it and then you have to press it that would create a new record saying that you have a new task and then you will get an notification above saying the task is added .Now once your task is added you will see that the task is added to the bottom of the tasks which are there if there is no task then your very first task it would be .In your php code you actually created a text field using a input tag where you will be taking input in the form of text now the name of that text input field is item and you will put a placeholder where you will write saying “Add A Item”. When you add a text saying the task that you will do then you will have a button just down the functionality of this button is that it is a submit button that means when you push the button what ever the data that you are passing through the field would be sent now the thing is that this submit button acts as a throwing field this will take input and throw it now we need to catch this data and you will do it when you will use isset function now this function will check there is such event which is done if yes then you will go in and you will then assign the item to an variable called as item or any thing that you like now what you will do is you will check whether the variable which is being assigned values is empty or not now if the variable is empty then there will be nothing to add but if at all there is something then what will happen is you will create another variable and now you will place a mysql query saying that you need to insert into todolist(name) values (‘$item’); now you will tell the compiler that you there is a table named todolist where you need to add the value which is present in the $item variable ,now where will you add this value the thing is that you will tell the compiler to add the data to the coloum which is coined as name which is actually a text field after that you will check wether the connection to the database is done or not if yes then your screen will have a alert msg saying that your message is successfully saved. But if your database connection is not connected then you will get a error saying that your connection with the database is terminated due to the not proper mention of the varibles or the syntax error now you task adding is done then. Once you add the data you will see that if there are data already present then your new task would be put at the end. If there is no task and this is your very first one then you will actually see that the icon where it showed that there is no tasks present will be gone and you will have a new task which you recently added .

Now once your task is created you will see that you will have two options saying that Mark as done and delete. Now The thing is that you will start doing the task then you will not be marking it has done but once you have done the task then you will click the button saying that the task is done.

Now before I go into that, there is a thing that I would like to say is that. Whenever you are adding a task. Each Task will actually have a separate item ID. Now each task will have two buttons which will be called as Mark as done and delete. Whenever you are pressing the button as Mark has done, what this PHP will do is it will send the parameters saying that action has done. Now The thing is that the PHP will catch the action. And. It will go to a function called as isset. Now the method that you're passing will be a get method. That means it doesn't have to be a unique. You do not have to worry about your data being leaked or something. Or the only thing that you're trying to send is that your your task has been done and you need to mark it. So what happens is the isset function will check whether the action is being called or not once you get the boolean value as true, then the method will run what happens into the matter is you will actually catch get method item and you will actually place that item value into the itemid variable. Then you will check if the action is equal to done then you will take another variable called as query and you are going to write this query which syntax which is "UPDATE todolist SET STATUS=1 WHERE id =$itemId'".After this we'll take an if condition where we'll be taking Now you will have a if condition in which you will actually run a function called as mysqli\_query which contains 2 parameters, one is the connection and another one is a query. Now the connection variable will be having four variables which is a server, a username, a password and a database. The server will actually have the server side URL which we're in this case is the local host. Then you will have your user name of the database there in which you are storing. Then you will be having the password. And the name of the database in which you are storing the data. And once your. Mysqli\_query is checking but all the boolean when it returns a boolean value and if at all the boolean value is true then it will go into the if condition or else it will go into the else block. Now in the else block we actually are printing the error which you are getting i'm getting in the connection. It will say you what kind of an error you're getting and where you're getting then. If at all the boolean value is true, then you will enter into the if condition where you will actually be getting an alert message saying that the item that you have marked as done is saved and the status has been changed to 1 once you mark the item as then then you will see that the item on which you have on which you have clicked the button as a mark as done. You will get a strike line from the item ID till the item now the same thing is happening with the delete button also. The moment you press the delete button your PHP will be passing 2 parameters which is action and an item. Now the this is thrown to the asset condition where you will be taking the action. And after that if the boolean value is true then it will go inside the if condition and it will. Uh, the item value that you are sending will be stored into an item ID variable. After that we have another if condition inside it which way you will be checking whether the action that you're sending is it done or a delete. If at all it is done then it will go to. But then function or else it will go into the delete function. Now after checking whether the action is delete or are done you will create a variable called as query and you will be passing a syntax of the MySQL database saying that "DELETE from todolist where id='$itemid'" where you are telling the database to delete that item at the particular item ID. After that you will taking another if condition. Now in this if condition you will again take a mysqli\_query function where you will be passing the two parameters such as connection and query. Now what this function does is actually it checks whether the connection is being formed or not. Once the connection is formed then it will send the query to that connection and it will Execute the query after that it will move into the if condition or else if the if at all there is some kind of an error in the MySQL function then it will go into the else block where you will be shown what kind of an error front end where you will be adding the task and another one will be at the back end or the database.there it is or where it has happened. In the if condition we have actually given output called us item which you have marked as delete is successfully deleted from the database. You can even go and check whether the item has been deleted or not and you will see that the item task has been deleted from two places one will be your. Database where you are storing all the tasks and its ID and status and another one will be the front end where you will be actually taking each and every input input from the database and you're presenting it to the user. Now whenever you add an item or delete an item or mark any item as I've done, you will get an alert message saying that the item has been added or deleted or mark has done. Now what is behind the function you will create a script function where you will be sending a document and you will be reading that. Function. Now what you're going to do is you're going to say that the message that you want to show it should be in alert and it should fade in about 5 seconds and after fading it should move on to the top. That means it should vanish completely going on to the top.

DataBase :-

Database is nothing but the all the data that you're saving that you're adding at the. Whatever may be that you're adding the task, you're deleting the task, you're marking the task as done. Whatever you're doing, it will be stored in the database where you. If at all you are adding a task, then you will have three to four attributes in which the first will be the task. Description the status of the task saying that whether it is in progress or whether it is being done or is it done. Then you will have the time stamp saying that when you created and when you updated it, then you will have the ID. As I told you before, each and every task has an item ID which is unique. And it is not enough then once for example if you made any changes saying that you have marked the item as done, then your status will be changed. That means you have updated that task at a particular time that is also even saved into this database. Now the main database language that we're using is MySQL the reason why we use MySQL is because it is an open source language and it also contains the relational database management system that enables the user to store, manage and reduce structured data efficiently. structured means that you will be saving the data in the form of tables and rows it is widely used for. Various applications from small scale projects to a very large scale websites and enterprises solutions.

Complete Working from the start till the end:-

In order to start this application, what we need to do is we need to have are Two things. One will be the xampp which is actually a cross-platform web server which helps developers to create and test their programs on a local web server. It was actually developed by the Apache friends and is a native source code can be revised or modified by the audience. It consists of Apache HTTP Server, Mariadb and Interpret. We're just like SPHP Pearl. It is also available in 11 languages and supported by different platforms such as IA-32 package of Windows & x64 package of macOS and Linux. Now once you have downloaded this application, what you're going to do is you are simply going to run it. You will actually. Start both the Apache and the MySQL server where you will be given two ports on which both the Apache and MySQL will be running, but they will not be running onto the same ports but on the different port. Once you have launched it, what you're going to do is you're going to yours. Your web browser. Then you're going on to the search bar and you're going to simply type “localhost:/8000” which is actually the common port on which you are going to run your. web application. Then you're going to type and the name by which you want to actually call this application. For me I have. In this I have put to todo list. Then when I'm running I will be getting a login page asking me the details of the e-mail and password then it will have a login. Button. Then you if you are a person who is known to this application then you can have a sign up uh button where you will be asking 3 fields which will be the. The very first pair will be the entire e-mail. Create a password, confirm your password. Now what it is going to happen is once you enter the e-mail. Yeah. Now when you give your e-mail and click on login you will be actually going on to the next page which is saying that you have the particular e-mail. Now you will have a link called To Do List. Once I click on that you will be directed to their particular. Application. Uh page where you will have this where UI saying that it is a two do list. You will have a field Goal is add an item if at all if you have any tasks before for that particular. Um. E-mail ID you will be getting them or else you can just type in task that you want. Then if you enter you can see that the alert message is saying that the item has been added successfully and when you see under the add item box you will see there the task description that you have. Event and just next to that you have you'll have two buttons saying as Mark has done and delete. Now the moment you type mark has done. What will happen is it will strike off that test description and you will be getting an alert box saying that the item has been marked. Then if at all you delete it, it will the complete task has been removed and you will get an alert. Saying that the item has been deleted successfully. No, if at all. If you were to go to the database then. You have to go to the particular database there on which you are running this. Then you will be seeing that you have two tables. The very first one is called a signup table which will actually take all the inputs from your e-mail and the second is the task description, the status and the other things if you go into sign up you will be having. Three uh. Attributes which will be sign number, your e-mail and the password that you're setting to that particular e-mail. Now in this you will observe that the e-mail that you're trying to send is very unique. You cannot put the same e-mail ID twice or thrice and the password can be or cannot be same. Then when you go back. To the second table. You'll be having. 5. Attributes now for. For example, I have created an IRA. Tas saying that uh, bring tomatoes is my. Tax, depreciation and when I add I'll be getting an ID 's ID number which is 1. Now the status is 0 and it will show me the time on which at which I have created this particular task and have I updated it. It will also give me the same task as been I've created. Now if I were to mark this particular task as done and when I again check in my database then we will see that the. Then I will see that the status has been marked as one and then I will see that the updated value is also done. And when if you were deleted have you want to delete that particular task then all you have to do is click on the button saying delete and it will give you the. Alert saying the data has been deleted successfully and when you run the refresh the database, you're going to see that the item that you have actually added will be removed from that database. Now let us see the for example. When you are adding some items, you will add it. After that, for example, when you're marking the item as done, if you can see it your search bar, you can see that you will be having the local host slash the table the file name in which you are actually running this. Then you will have something. Called as the action is equals to done and item is 4 Nida item. Item is actually the ID on which you are marking the task as done and the action which is your passing is. It can be two types. It can be either done or delete. So. According to that your task will be either deleted or it will be marked as done.

Code :-

Login Page.php:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <title>Login & Registration Form</title>

  <style>

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Poppins', sans-serif;

}

body{

  min-height: 100vh;

  width: 100%;

  background: #009579;

}

.container{

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%,-50%);

  max-width: 430px;

  width: 100%;

  background: #fff;

  border-radius: 7px;

  box-shadow: 0 5px 10px rgba(0,0,0,0.3);

}

.container .registration{

  display: none;

}

#check:checked ~ .registration{

  display: block;

}

#check:checked ~ .login{

  display: none;

}

#check{

  display: none;

}

.container .form{

  padding: 2rem;

}

.form header{

  font-size: 2rem;

  font-weight: 500;

  text-align: center;

  margin-bottom: 1.5rem;

}

 .form input{

   height: 60px;

   width: 100%;

   padding: 0 15px;

   font-size: 17px;

   margin-bottom: 1.3rem;

   border: 1px solid #ddd;

   border-radius: 6px;

   outline: none;

 }

 .form input:focus{

   box-shadow: 0 1px 0 rgba(0,0,0,0.2);

 }

.form a{

  font-size: 16px;

  color: #009579;

  text-decoration: none;

}

.form a:hover{

  text-decoration: underline;

}

.form input.button{

  color: #fff;

  background: #009579;

  font-size: 1.2rem;

  font-weight: 500;

  letter-spacing: 1px;

  margin-top: 1.7rem;

  cursor: pointer;

  transition: 0.4s;

}

.form input.button:hover{

  background: #006653;

}

.signup{

  font-size: 17px;

  text-align: center;

}

.signup label{

  color: #009579;

  cursor: pointer;

}

.signup label:hover{

  text-decoration: underline;

}

  </style>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container">

    <input type="checkbox" id="check">

    <div class="login form">

      <header>Login</header>

      <form action="verification.php" method="post">

        <input type="email" name="email" placeholder="Enter your email">

        <input type="password" name="password" placeholder="Enter your password">

        <a href="?">Forgot password?</a>

        <input type="submit" class="button" value="Login">

      </form>

      <div class="signup">

        <span class="signup">Don't have an account?

         <label for="check">Signup</label>

        </span>

      </div>

    </div>

    <div class="registration form">

      <header>Signup</header>

      <form action="firsttime.php" method="post">

        <input type="email" name="email" placeholder="Enter your email">

        <input type="password" name="password" placeholder="Create a password">

        <input type="password" name="currentpassword" placeholder="Confirm your password">

        <input type="submit" class="button" value="Signup">

      </form>

      <div class="signup">

        <span class="signup">Already have an account?

         <label for="check">Login</label>

        </span>

      </div>

    </div>

  </div>

</body>

</html>

Verification.php

<html>

    <head>

        <title>Email and password verification</title>

    </head>

    <body>

    <?php

    $server = 'localhost';

    $username = 'root';

    $password = '';

    $database = 'to\_do\_list';

    $connection = mysqli\_connect($server,$username,$password,$database);

    if($connection->connect\_error){

        die('Connection to my sql failed '.$connection->connect\_error);

    }

    $email=$\_POST['email'];

    $password1=$\_POST['password'];

    $query ="SELECT \* from signup";

    $result=mysqli\_query($connection,$query);

    $flag=0;

    while( $row = $result->fetch\_assoc()) {

        $email1=$row['email'];

        $password2=$row['pass'];

        if($email==$email1 && $password1==$password2){

            $flag=1;

            break;

            ;

        }

        else{

            $flag=0;

        }

    }

    if($flag==1){

        echo '

            <center>

                        <div class="alert alert-success" role="alert">

                            Done

                            <h1><a href="todolist.php">To Do List</a> </h1>

                        </div>

                    </center>';

    }

    else{

        echo '

        <center>

                        <div class="alert alert-danger" role="alert">

                            This email already exists or you havent created it yet so go back thank you

                            <h1><a href="index.php">Go Back</a> </h1>

                        </div>

                    </center>

        ';

    }

?>

    </body>

</html>

FirstTime.php

<?php

    $server = 'localhost';

    $username = 'root';

    $password = '';

    $database = 'to\_do\_list';

    $connection = mysqli\_connect($server,$username,$password,$database);

    if($connection->connect\_error){

        die('Connection to my sql failed '.$connection->connect\_error);

    }

    $email=$\_POST['email'];

    $password1=$\_POST['password'];

    $password2=$\_POST['currentpassword'];

    if($password1 == $password2){

        $query ="INSERT INTO signup values(NULL,'$email','$password1')";

        $result=mysqli\_query($connection,$query);

        echo '<center>

        <div class="alert alert-success" role="alert">

            Go back and login again

            <h1><a href="index.php">To Do List</a> </h1>

        </div>

    </center>

';

    }else{

        echo '

        <center>

        Please Give Both the passwords same

        <a href="index.php">Link</a>

        </center>

        ';

    }

?>

ToDoList.php

<?php

    $server = 'localhost';

    $username = 'root';

    $password = '';

    $database = 'to\_do\_list';

    $connection = mysqli\_connect($server,$username,$password,$database);

    if($connection->connect\_error){

        die('Connection to my sql failed '.$connection->connect\_error);

    }

    //creating todo item

    if(isset($\_POST['add'])){

        $item=$\_POST['item'];

        if(!empty($item)){

            $query ="INSERT INTO todolist(name) values ('$item')";

            if(mysqli\_query($connection,$query)){

                echo '

               <center>

               <div class="alert alert-success" role="alert">

               Item Added Successfully!

           </div>

               </center>

                ';

            }else{

                echo mysqli\_error($connection);

            }

        }

    }

    //checking if action parameter is present

    if(isset($\_GET['action'])){

        $itemid=$\_GET['item'];

        if($\_GET['action']=='done'){

            $query ="UPDATE todolist SET status=1 where id ='$itemid'";

            if(mysqli\_query($connection,$query)){

                echo '

                <center>

                    <div class="alert alert-info" role="alert">

                        Item Marked as Done!

                    </div>

                </center>

                ';

            }else{

                echo mysqli\_error($connection);

            }

        }

    }

    //Deleting the Operations

    if(isset($\_GET['action'])){

        $itemid=$\_GET['item'];

        if($\_GET['action']=='delete'){

            $query ="DELETE from todolist where id='$itemid'";

            if(mysqli\_query($connection,$query)){

                echo '

                <center>

                    <div class="alert alert-danger" role="alert">

                        Item Deleted Suc

                        cessfully!

                    </div>

                </center>

                ';

            }else{

                echo mysqli\_error($connection);

            }

        }

    }

?>

<!DOCTYPE html>

<html>

    <head>

        <title>To Do List Application</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

        <style>

            .done{

                text-decoration: line-through;

            }

        </style>

    </head>

    <body>

        <main>

            <div class="container pt-5">

                <div class="row">

                    <div class="col-sm-12 col-md-3"></div>

                    <div class="col-sm-12 col-md-6">

                        <div class="card">

                            <div class="card-header">

                                <p>TODO LIST</p>

                                <div class="card-body">

                                    <form method="POST" action ="<?= $\_SERVER['PHP\_SELF']?>">

                                        <div>

                                            <input type="text" class="form-control" name="item" placeholder="Add a Item"><br>

                                        </div>

                                        <input type="submit" class="btn btn-dark" name="add" value="Add Item">

                                    </form>

                                    <div class="mt-5 mb-5">

                                        <?php

                                            $query ="SELECT \* FROM todolist";

                                            $result=mysqli\_query($connection,$query);

                                            if($result->num\_rows > 0){

                                                $i=1;

                                                while( $row = $result->fetch\_assoc()) {

                                                    $done = $row['status']==1 ? "done":"";

                                                    echo '

                                                    <div class="row mt-4" >

                                                        <div class="col-sm-12 col-md-1"><h5>'.$i.'</h5></div>

                                                        <div class="col-sm-12 col-md-5"><h5 class="'.$done.'">'.$row['name'].'</h5></div>

                                                        <div class="col-sm-12 col-md-6">

                                                            <a href="?action=done&item='.$row['id'].'" class="btn btn-outline-dark">Mark as Done</a>

                                                            <a href="?action=delete&item='.$row['id'].'" class="btn btn-outline-danger">Delete</a>

                                                        </div>

                                                    </div>

                                                    ';

                                                    $i++;

                                                }

                                            }

                                            else{

                                                $query ="TRUNCATE TABLE todolist";

                                                mysqli\_query($connection, $query);

                                                echo '

                                                <center>

                                                <img src="emptyfolder.png" width="100px" alt="No tasks yet"><br><span><b>Your List is Empty</b></span>

                                                </center>

                                                ';

                                            }

                                        ?>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </main>

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

        <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

        <script>

            $(document).ready(function(){

                $(".alert").fadeTo(5000,500).slideUp(500,function(){

                    $(".alert").slideUp(5000,500);

                })

            })

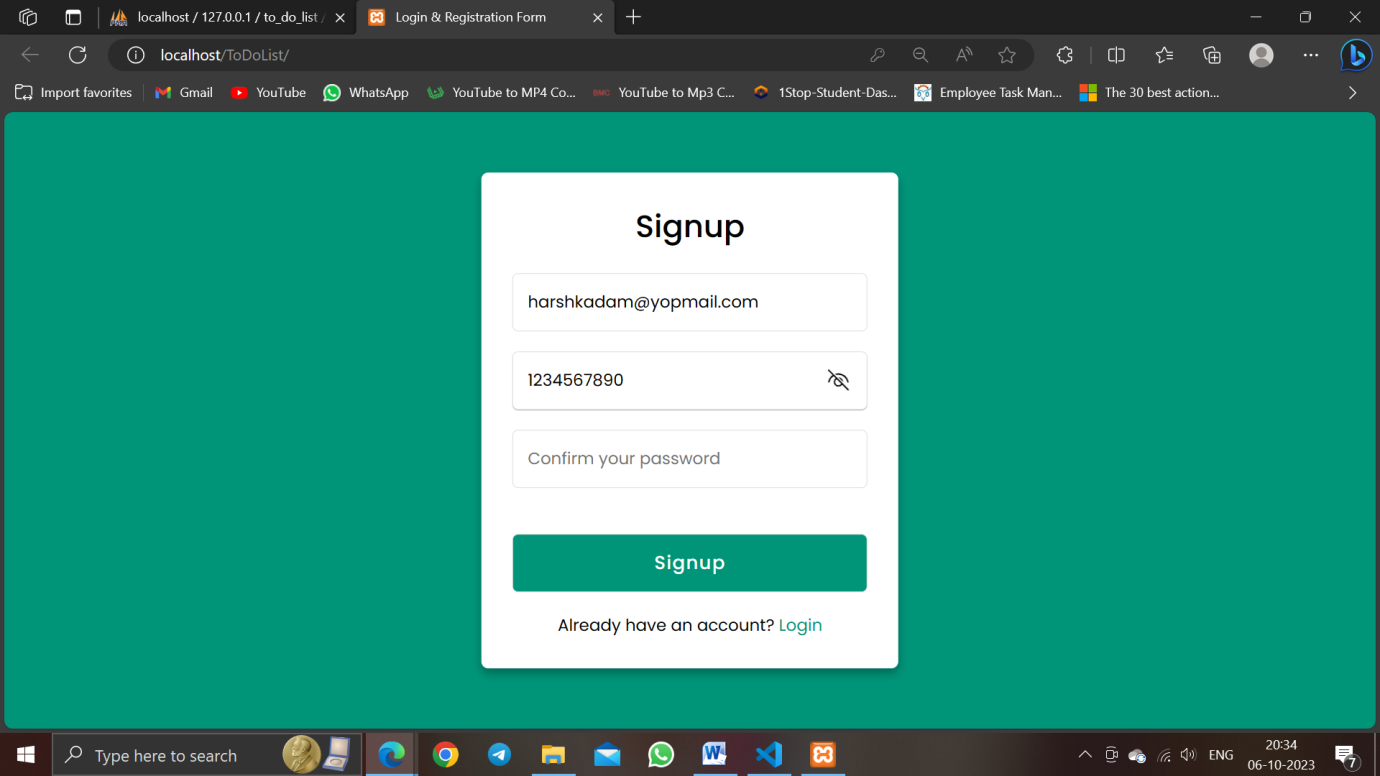
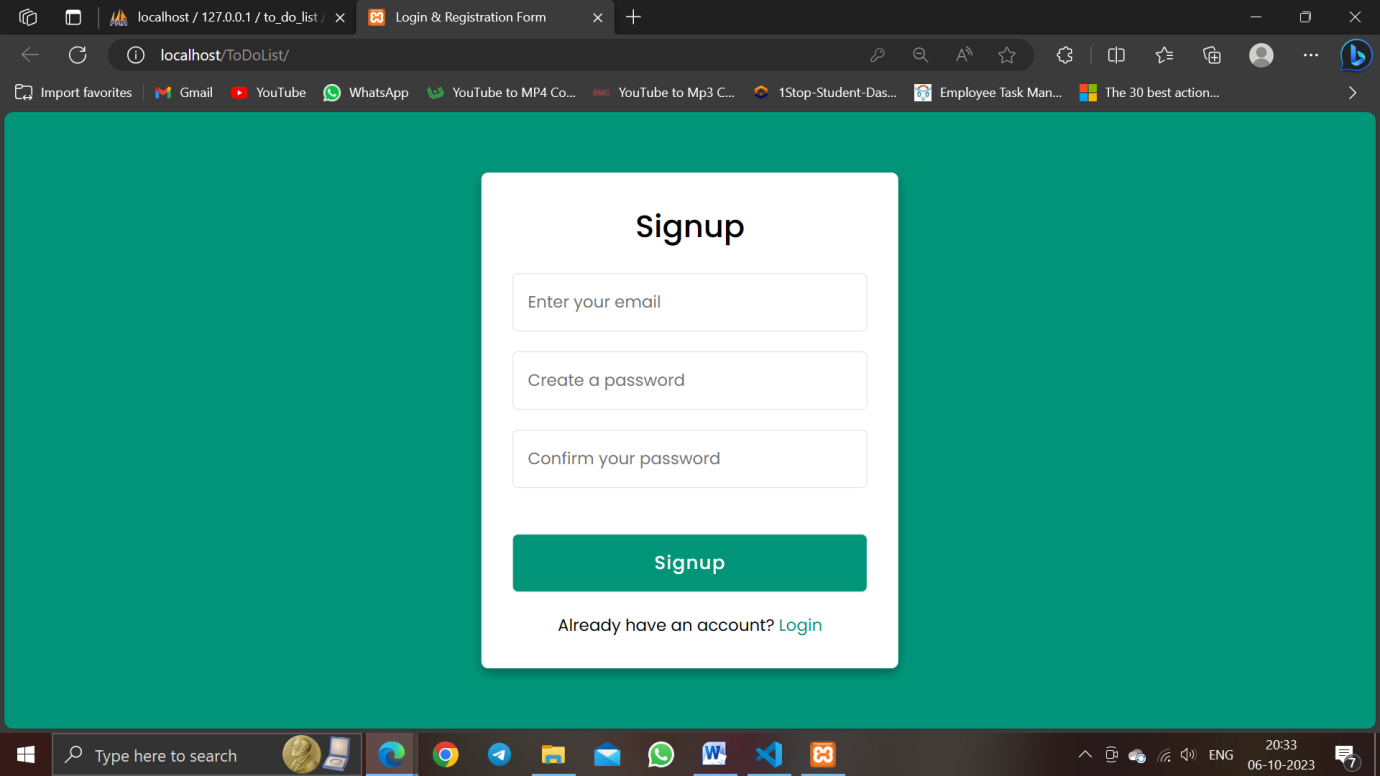
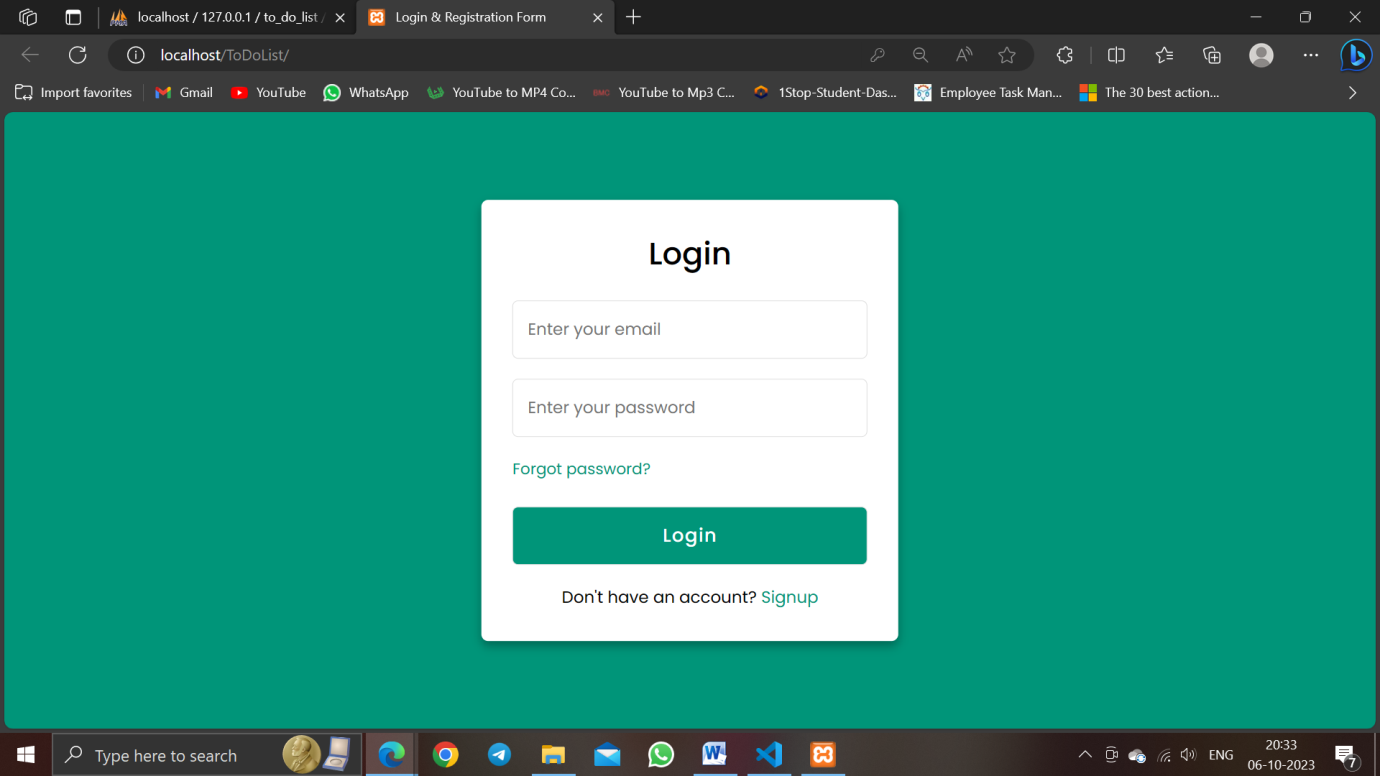
        </script>

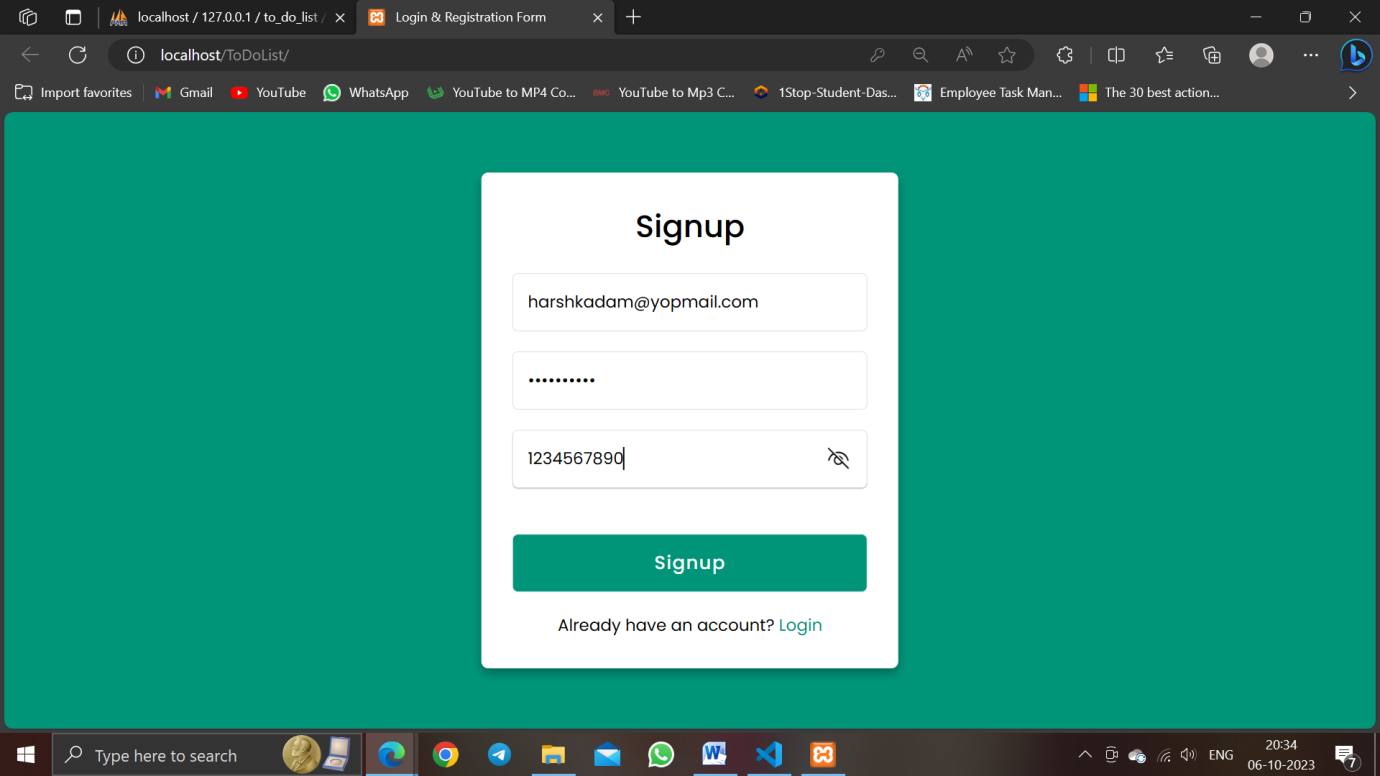
    </body>

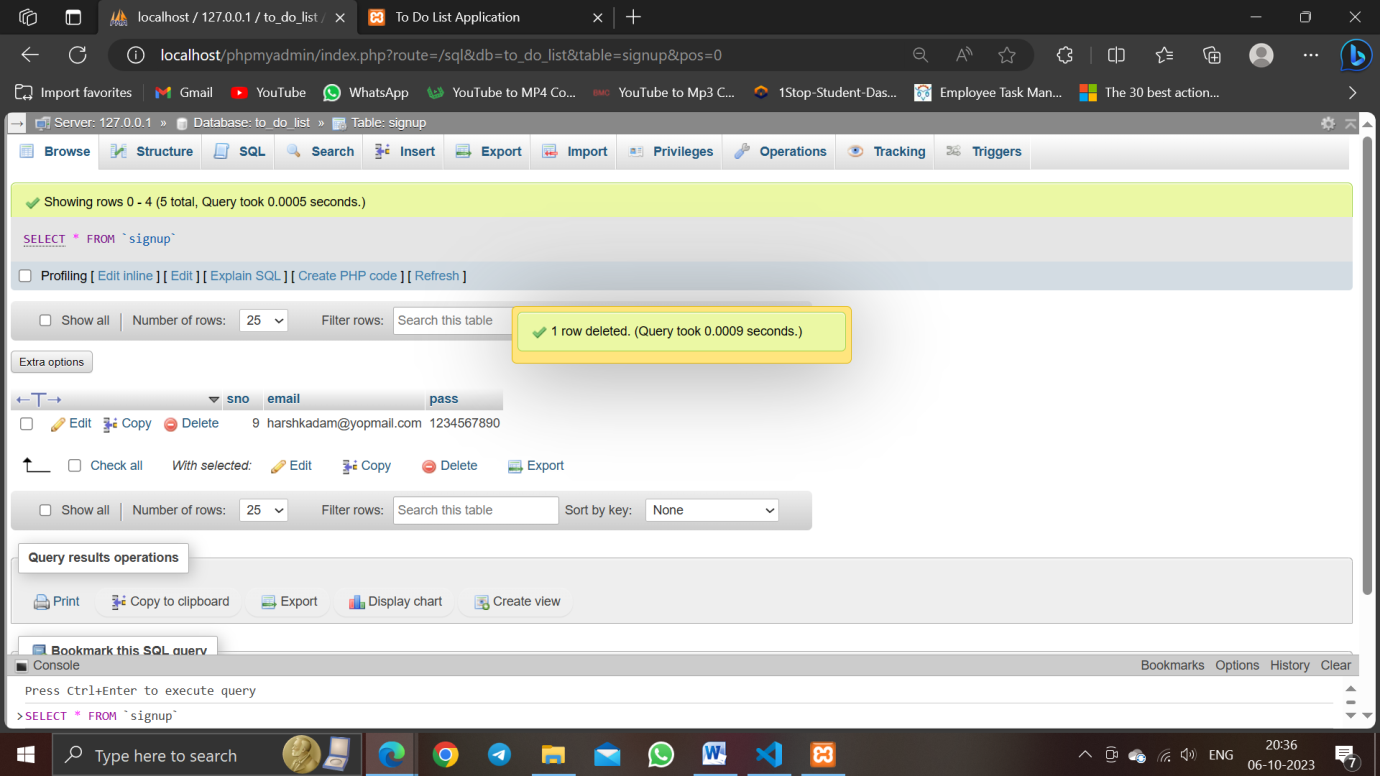
</html>

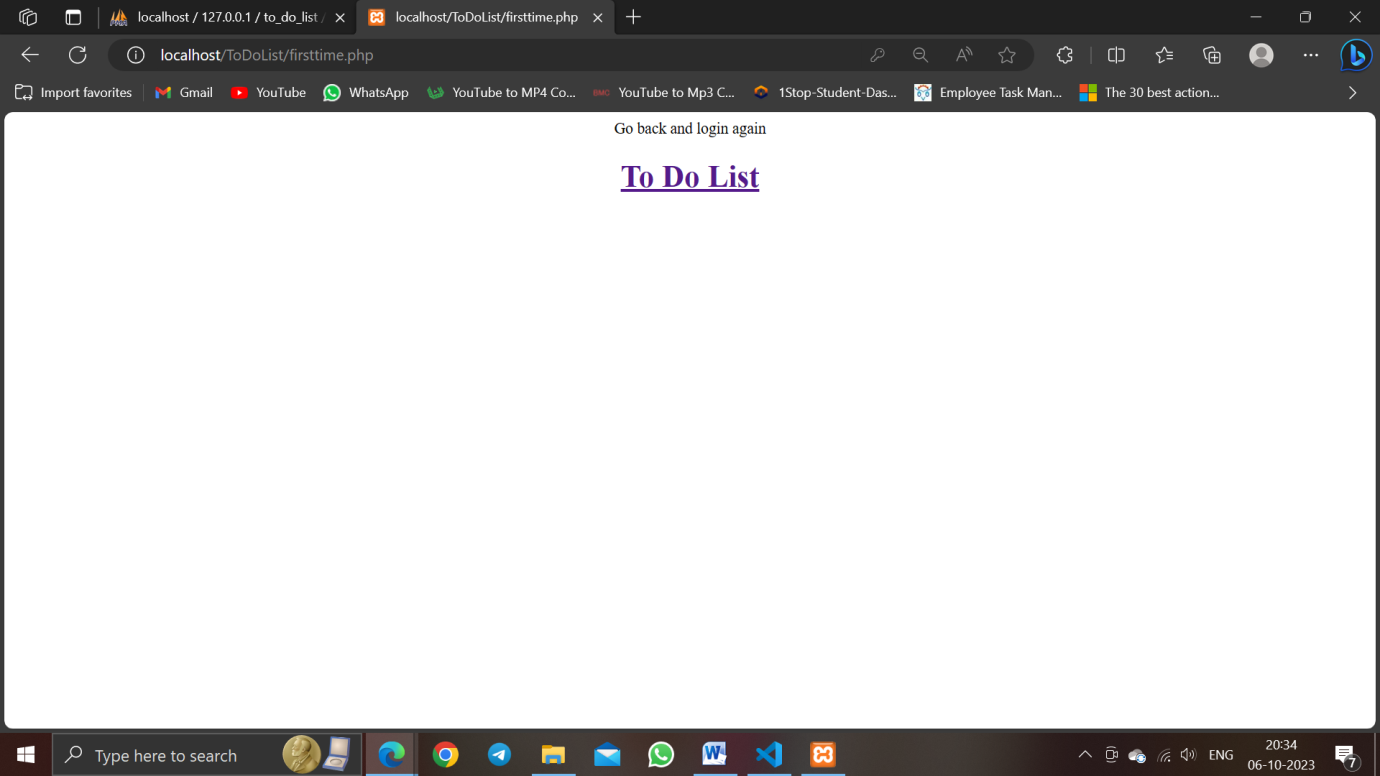
Outputs Related to the Project :

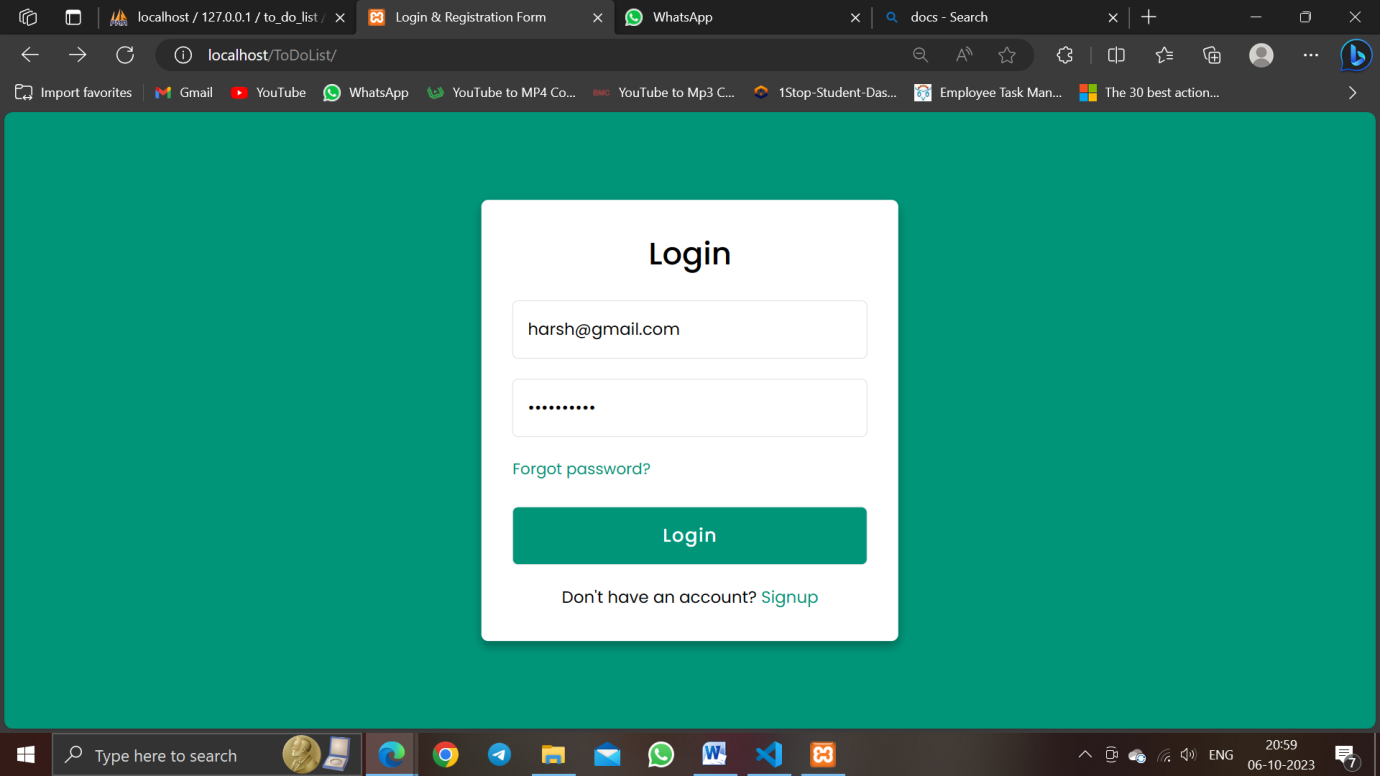
This is the Login Page :

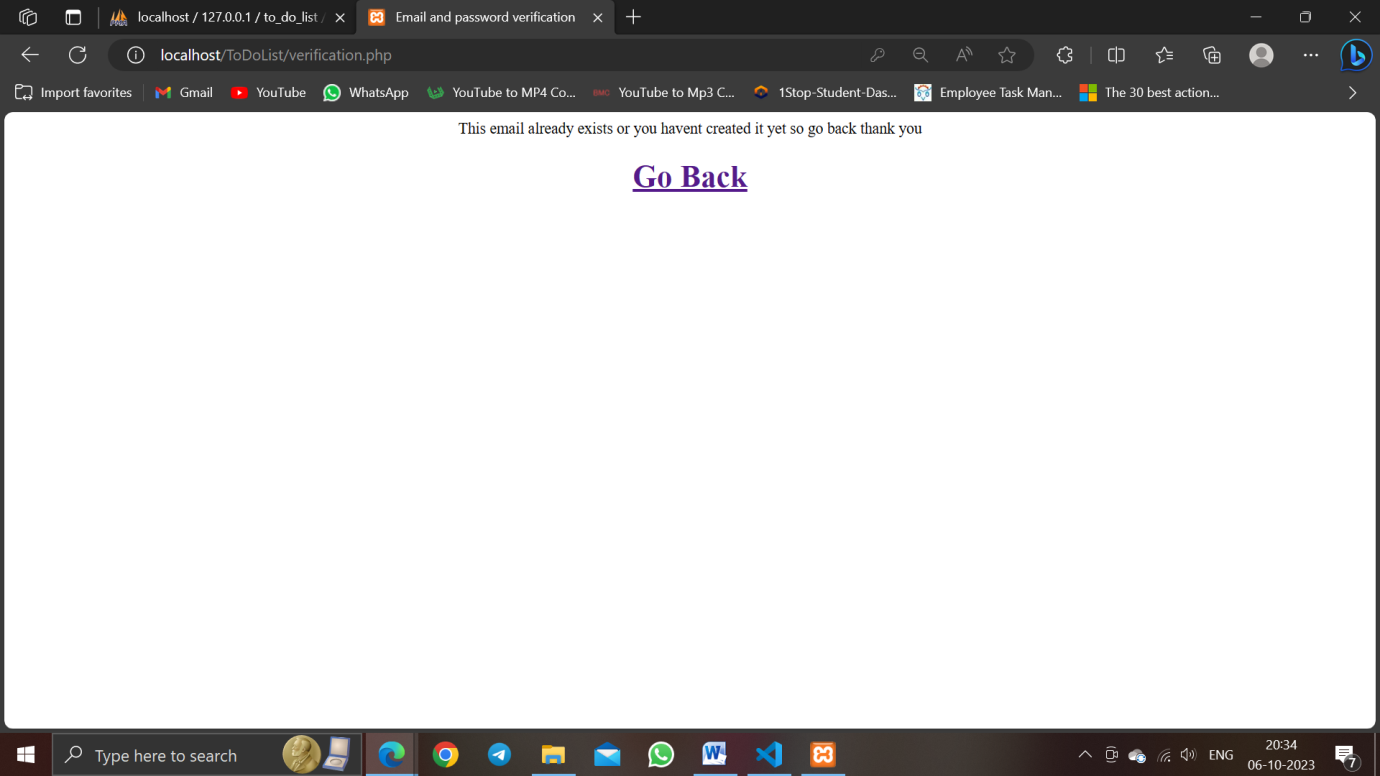


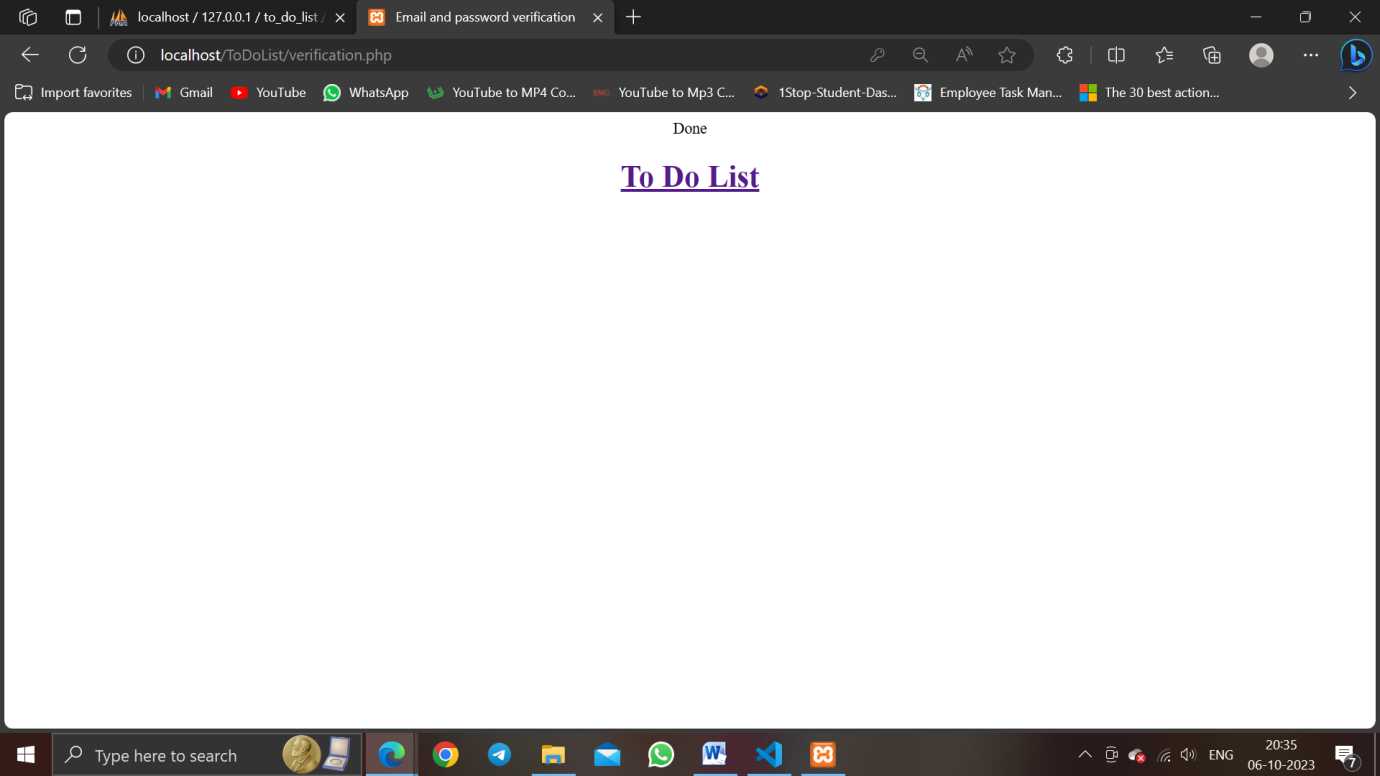
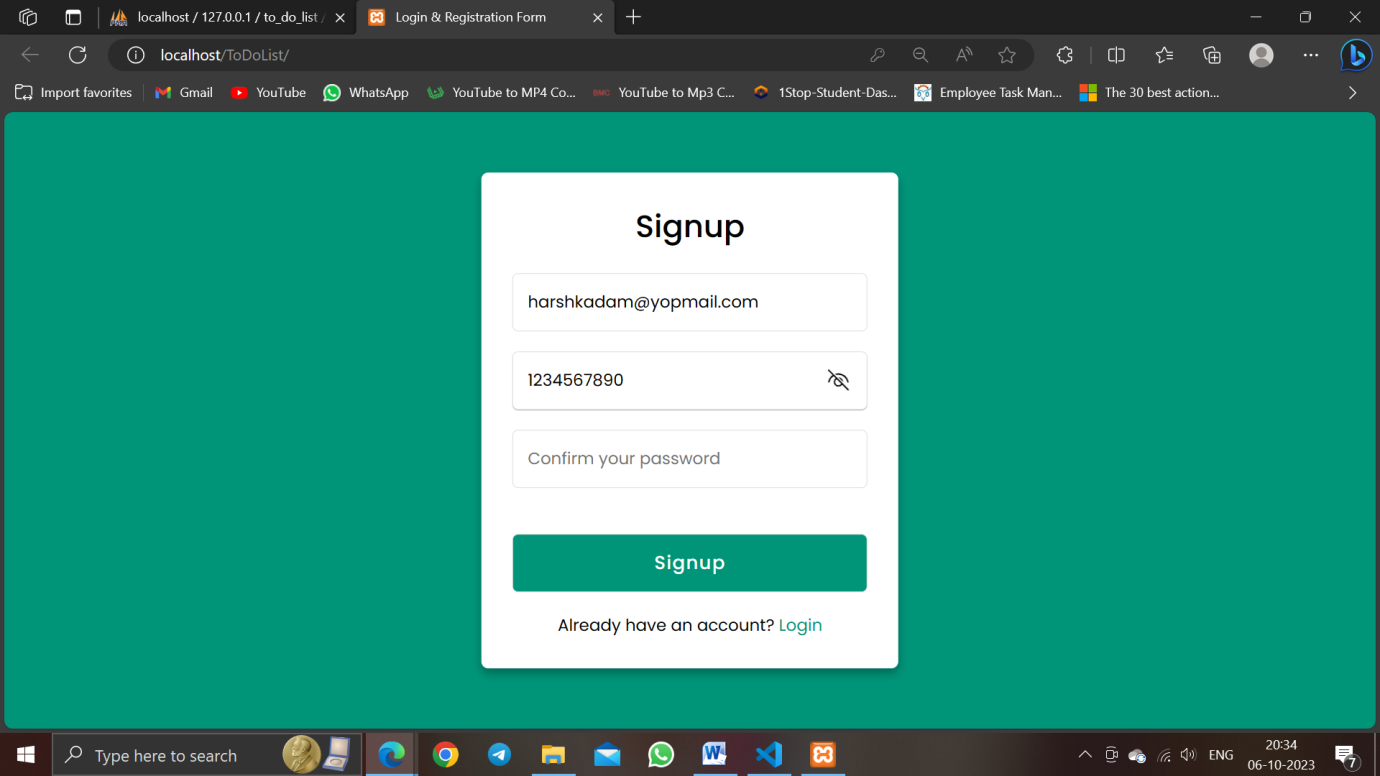




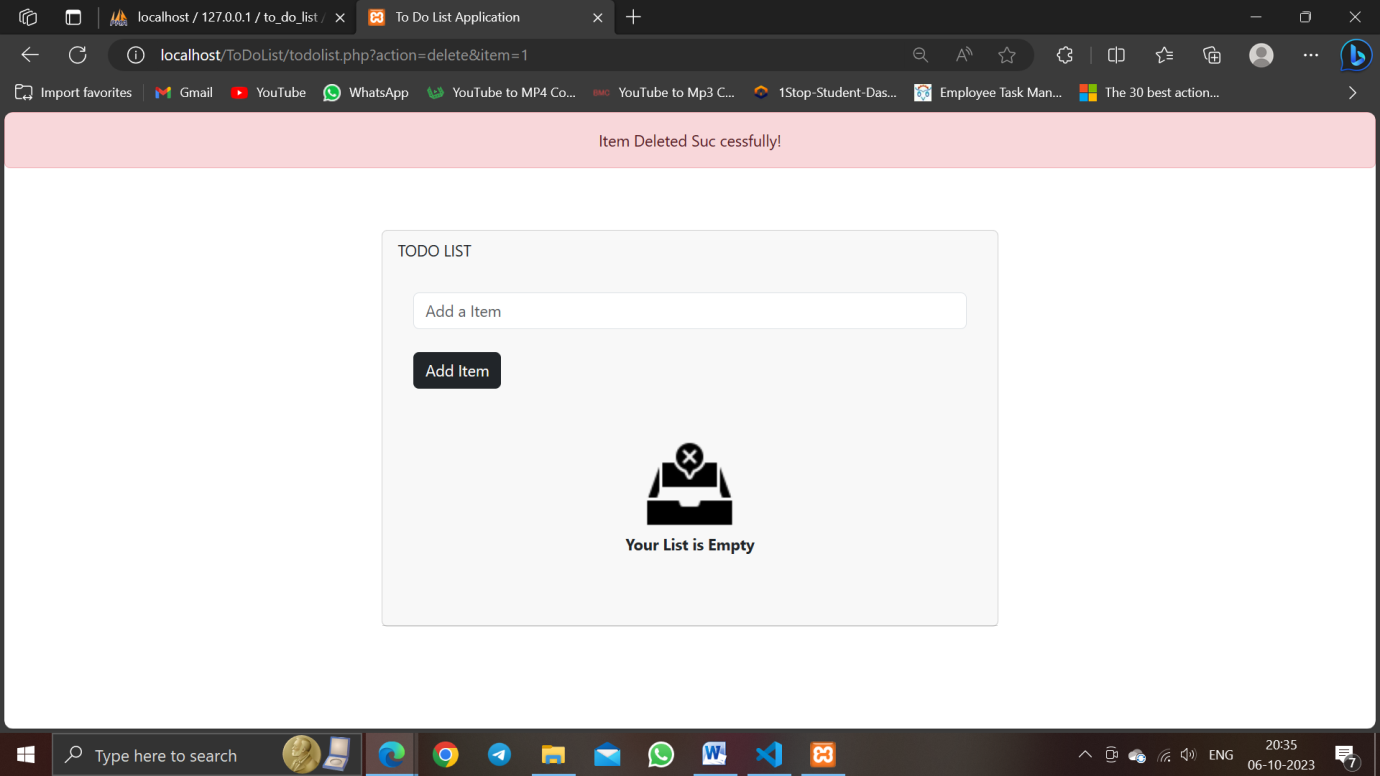
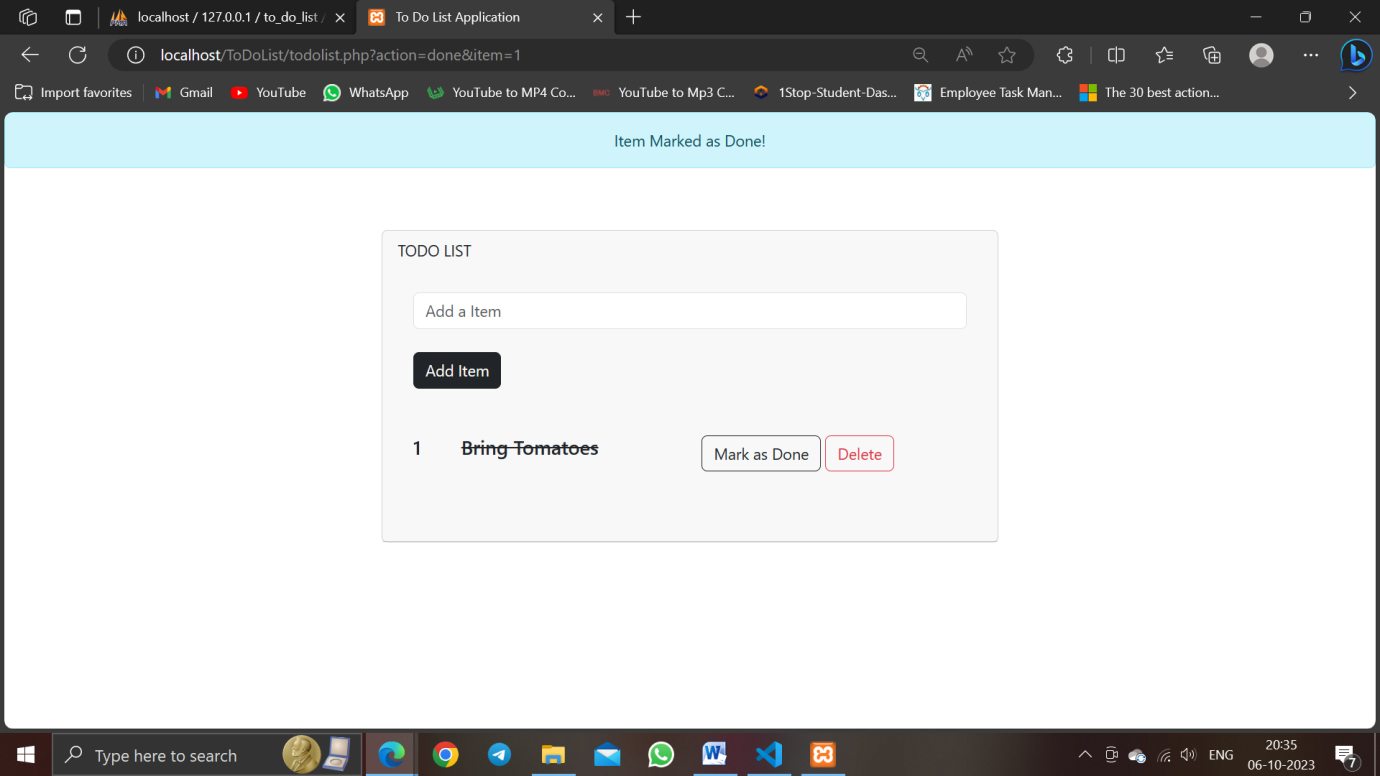
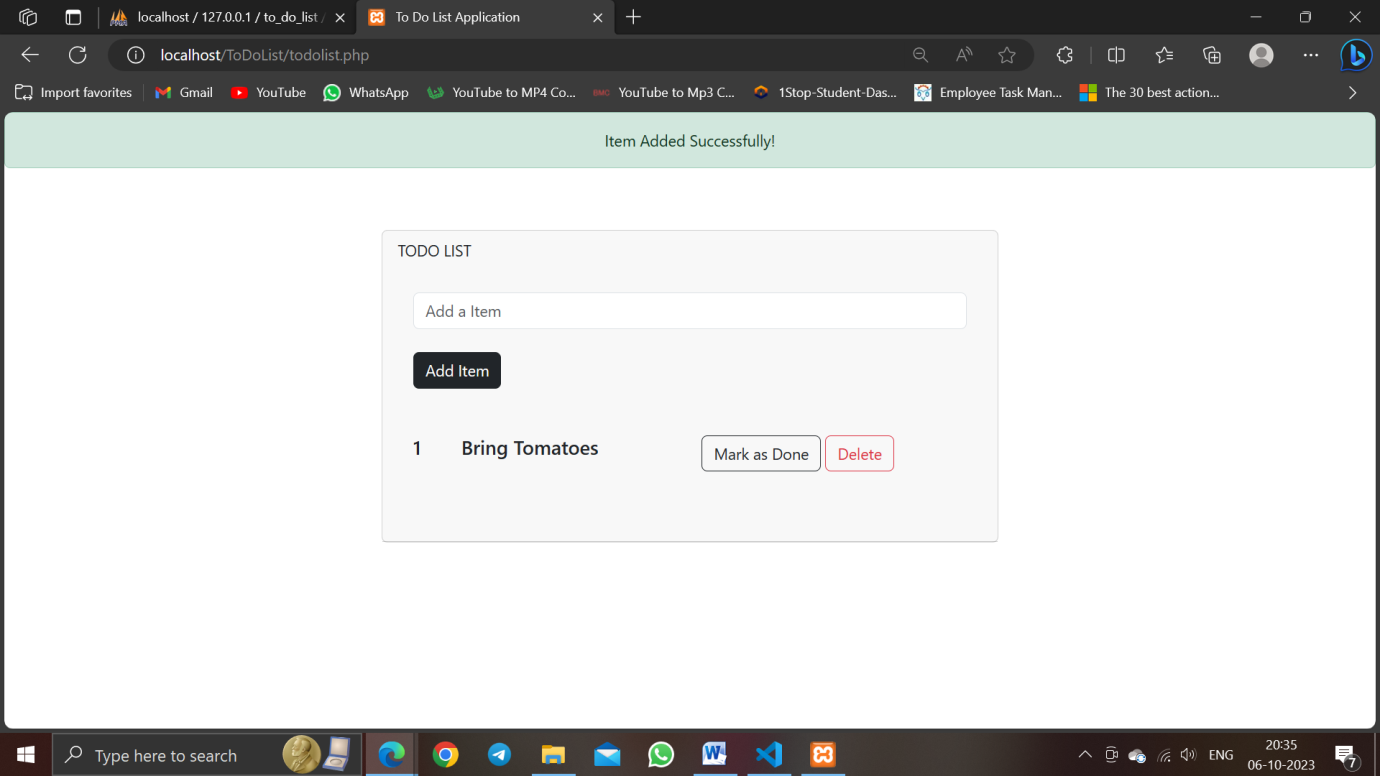
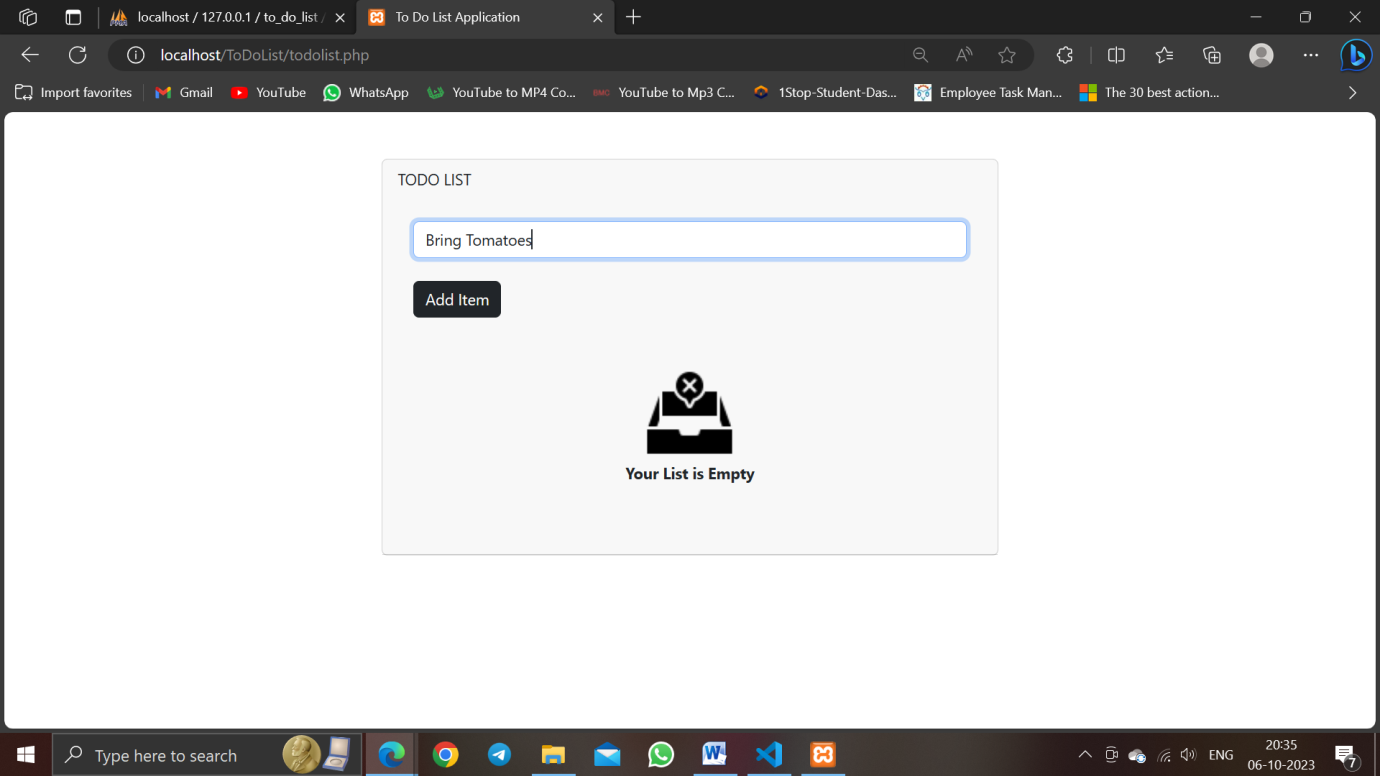
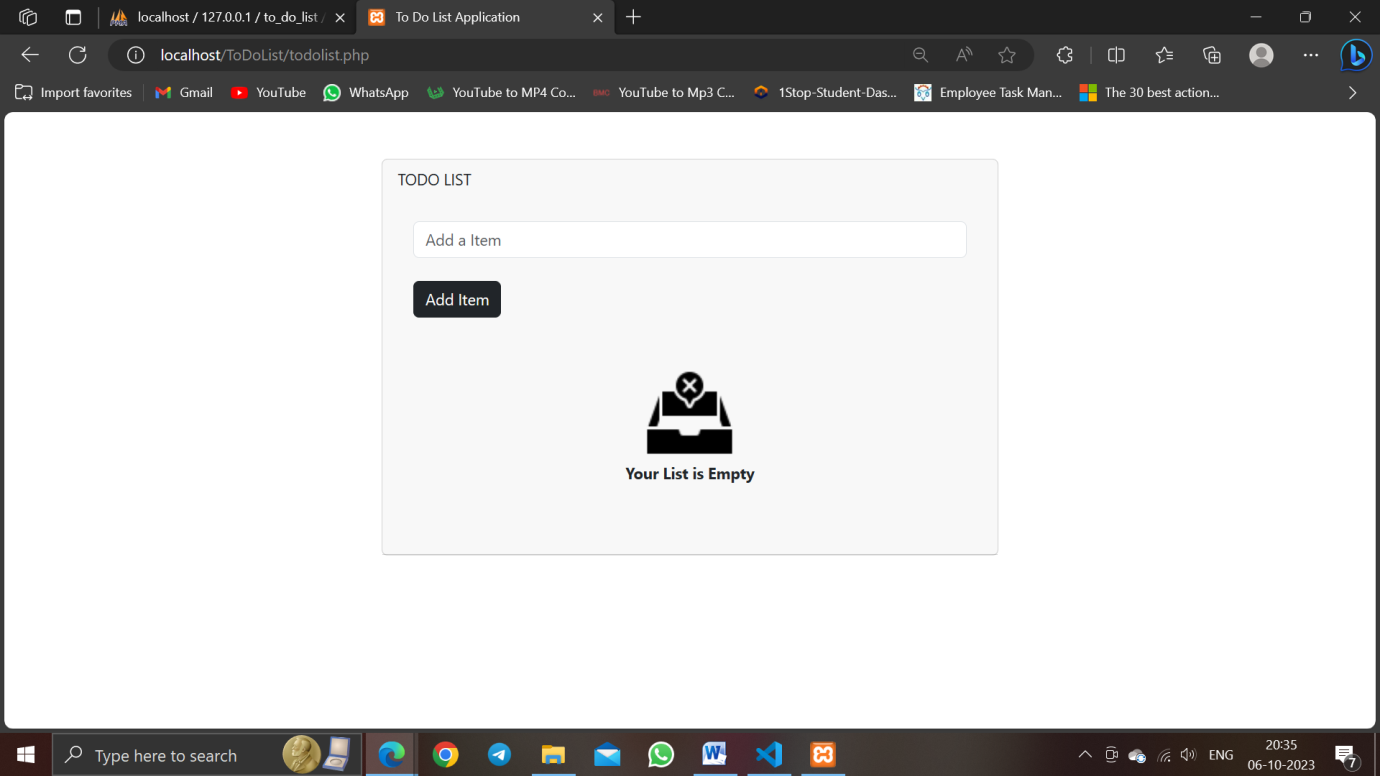




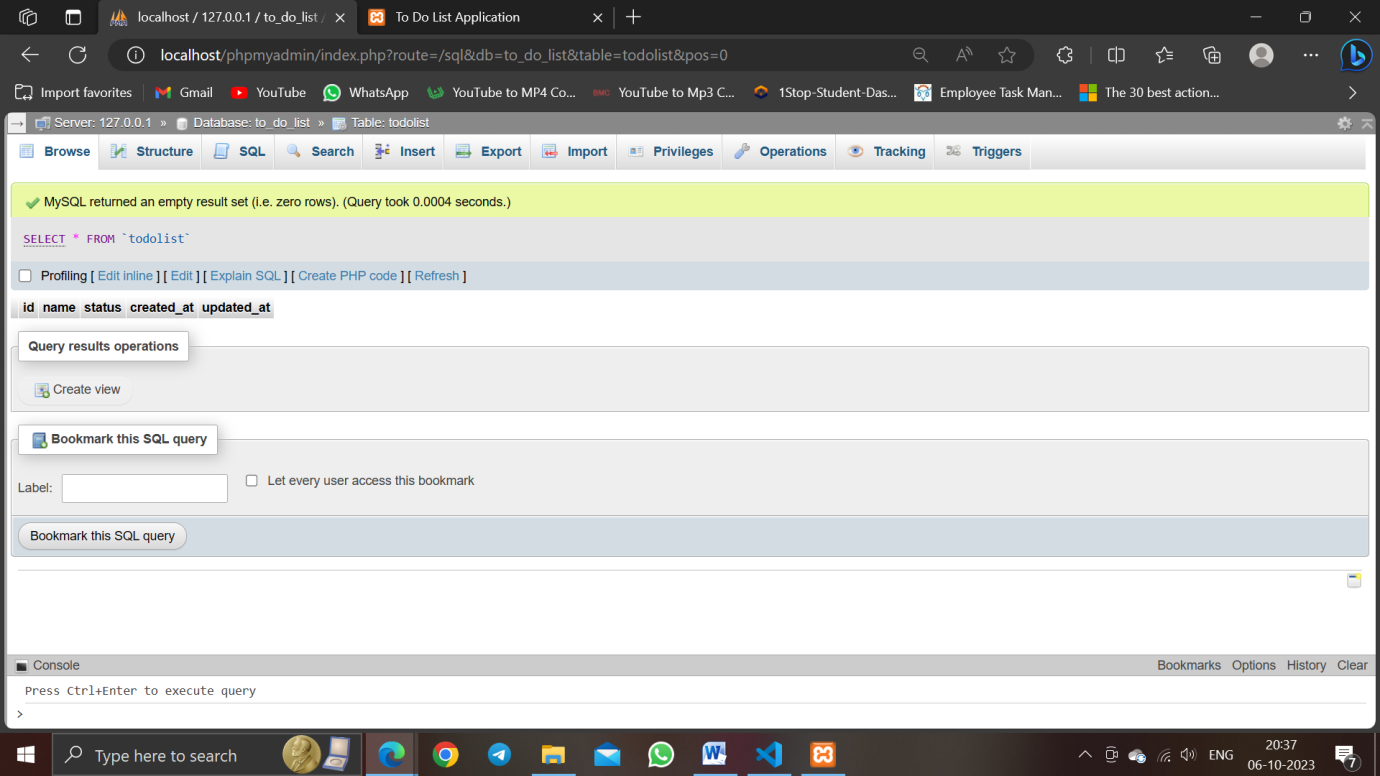
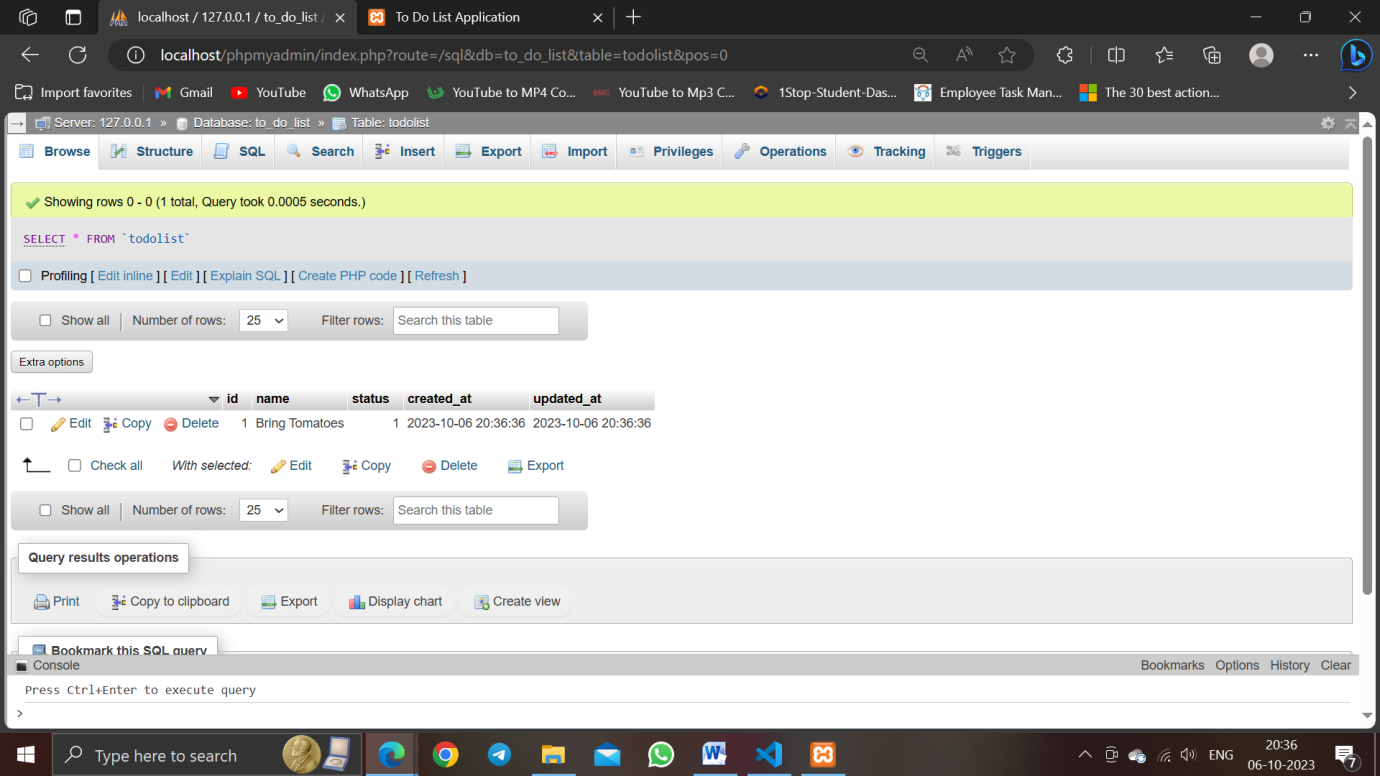
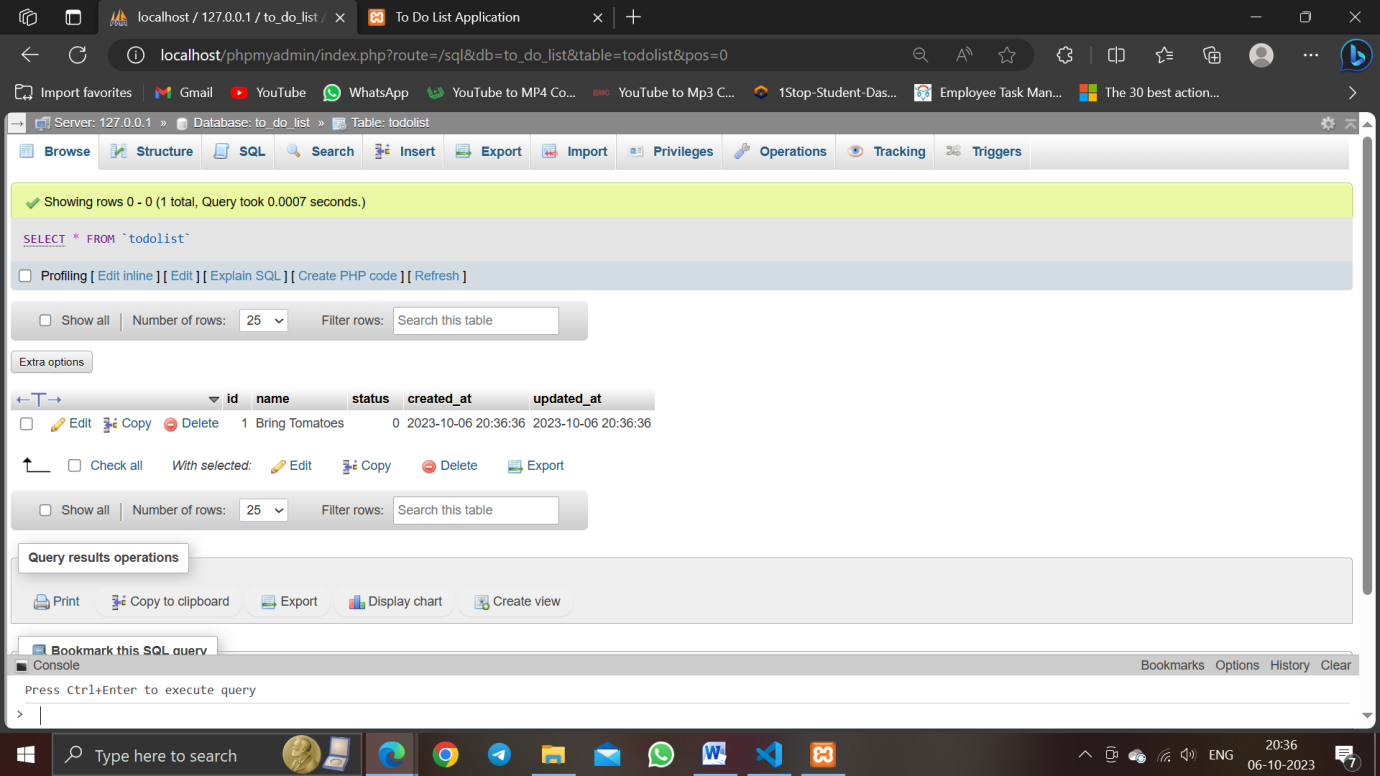
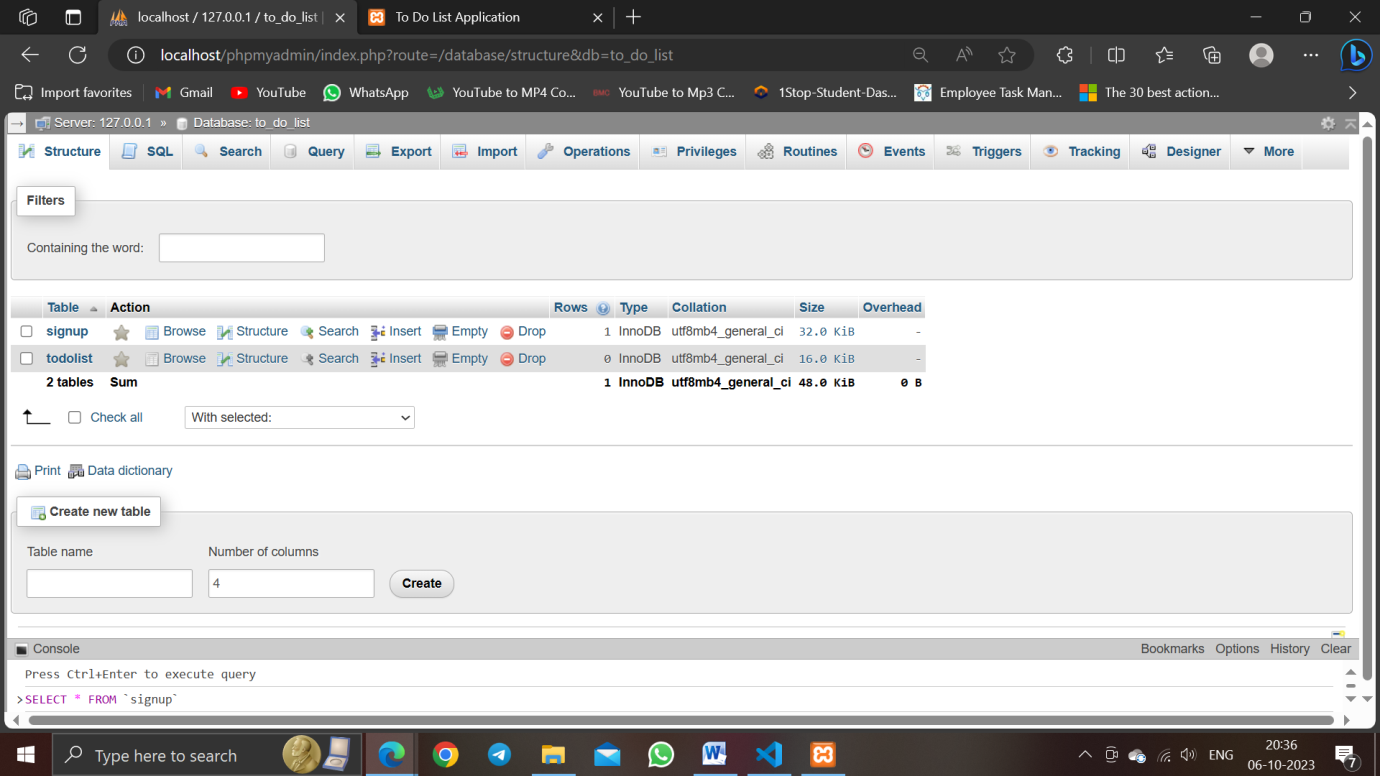




ToDoList:



DataBase:



***Conclusion:-***

To Conclude, About project I would like to say that the main reason why we built this application was so that the user can have a proper way of organizing his day to day tasks.

The main challenges that we faced in this project was that the proper indention of the HTML. I was facing issues as to how to properly align my code and then there was the problem with the linking of the html and php code.If I would like to add I would add a few more things where the user can prioritize his/her task using date and adding a email application where once the user adds or marks the task as done the application should send a email saying that the new task has been added or the task has been marked and so much more can we add.

Git Hub Link For the Project :-> [Projects-1/ToDoList at main · Harshkadam1217/Projects-1 (github.com)](https://github.com/Harshkadam1217/Projects-1/tree/main/ToDoList)

The video as how this project works is given in this project link you need to first clone it and then play it.