

CSE-190021

Design and Development an Online Daily Tasks Management System (My ToDo)

by

Shahed Masud Rabbi
ID No. CSE1601007050

Fateha Binte Amin
ID No. CSE1601007070

Fazle Rabbi
ID No. CSE1601007079

Parveen Akter
ID No. CSE1601007041

Salvin Islam
ID No. CSE 1601007056



A Project Submitted in Partial Fulfillment to the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

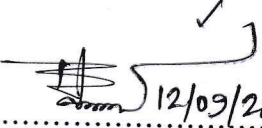

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

September 2019

APPROVAL

The Project Report **Design and Development an Online Daily Task Management System (My ToDo)** submitted by Shahed Masud Rabbi ID No. CSE1601007050, Fateha Binte Amin ID No. CSE1601007070, Fazle Rabbi ID No. CSE1601007079, Salvin Islam ID No. CSE1601007056, Parveen Akter ID No. CSE1601007041 to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

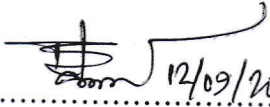
1. Bulbul Ahamed  12/09/2019 (Supervisor)
2. Arifur Rahaman Arif 12.09.2019 (Examiner)
3. Khadija Islam  12.09.19 (Examiner)

Bulbul Ahamed
Associate Professor & Head
Department of Computer Science & Engineering
Sonargaon University (SU)

DECLARATION


We, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of **Bulbul Ahamed**, Associate Professor and Head, Department of Computer Science and Engineering, Sonargaon University (SU). We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned


12/09/2019

(Bulbul Ahamed)
Supervisor


Signature



(Shahed Masud Rabbi)




(Fateha Binte Amin)



(Fazle Rabbi)



(Parveen Akter)



(Salvin Islam)

Candidates

ABSTRACT

A good to-do list web application is not just a dashboard of our life's obligations - it can be the control room that makes it possible to keep chaos at bay. Our web application software (My ToDo) has flexible organization schemes and thoughtful designs that make it easy to duck in, check on obligations, enter new tasks, and then get back to the doing. My ToDo web application includes a powerful system for logging and reporting time spent on tasks. Task lists can also be printed or saved in various other formats for display and reporting purposes. For developing this web application, we have used a variety of tools e.g. HTML, CSS, JavaScript, PHP, MySQL, XAMPP etc.

ACKNOWLEDGEMENTS

We are most grateful to almighty Allah for His loyal help, which is most effective tool for all kind of achievements of all time and enable us to complete the successfully.

We are also auspicious that we had the kind association as well as supervision of honorable supervisor **Bulbul Ahamed**, Associate Professor and Head, Department of Computer Science and Engineering, Sonargaon University (SU), for his continuous guidance, encouragement, and patience, and for giving us the opportunity to do this work. His valuable suggestions and strict guidance made it possible to prepare a well-organized report.

We would like to convey our special gratitude to **Prof. Dr. M.A. Mabud**, Dean, Faculty of Science & Engineering, Sonargaon University (SU), for his kind concern and valuable suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Last of all we are grateful to our family; who are always with us in every step of life.

TABLE OF CONTENTS

	<u>Page No</u>
Abstract -----	iii
Acknowledgements -----	iv
 Chapter 1: Introduction	
1.1 Introduction of the project -----	2
1.2 Objectives of the project -----	2
1.3 Project Overview Diagram -----	3
1.4 Descriptions of the project -----	3
1.5 Design and Implementation -----	6
 Chapter 2: Object-Oriented Design	
2.1 Introduction of Object-Oriented Design -----	8
2.2 Use Case Diagram -----	8
2.3 Activity Diagram -----	10
2.4 Class Diagram -----	12
2.5 Entity Relationship Diagram -----	13
 Chapter 3: User Interface Design	
3.1 Introduction of User Interface Design -----	16
 Chapter 4: Conclusion and Future Work	
4.1 Conclusion -----	22
4.2 Future Work -----	22
References -----	23
Appendix -----	25

CHAPTER -1
INTERODUCTION

1.1 Introduction of the Project

Most of the people have many more activities on their “wish lists” than the time available to work on them. By choosing activities intelligently, they can make the very most of their time and opportunities. However, by choosing badly, they can bog themselves down in time-depleting, low-yield projects that stop us moving forward. This is where an “Online Daily Tasks Management System (My ToDo)” can be useful. This helps us to choose the activities we should prioritize and the ones we should avoid if we want to make the most of our time and opportunities.

1.2 Objectives of the project

Three major objectives are listed below-

- **Organization:** Organizing your tasks with a list can make everything much more manageable and make you feel grounded. Seeing a clear outline of your completed and uncompleted tasks will help you feel organized and stay mentally focused.
- **Improved Memory:** Keeping this “Online Task Management System” improves your memory by giving you permission to forget. Feeling forgetful? Nobody is capable of remembering everything at all times. To-do lists are a useful external memory aid that give you permission to forget. As long as you can remember to look at your to-do list, you will never lose anything that you have recorded on it.
- **Productivity:** If you record all your tasks in a to-do list, you can easily review the list and prioritize the most important tasks.

1.3 Project Overview Diagram

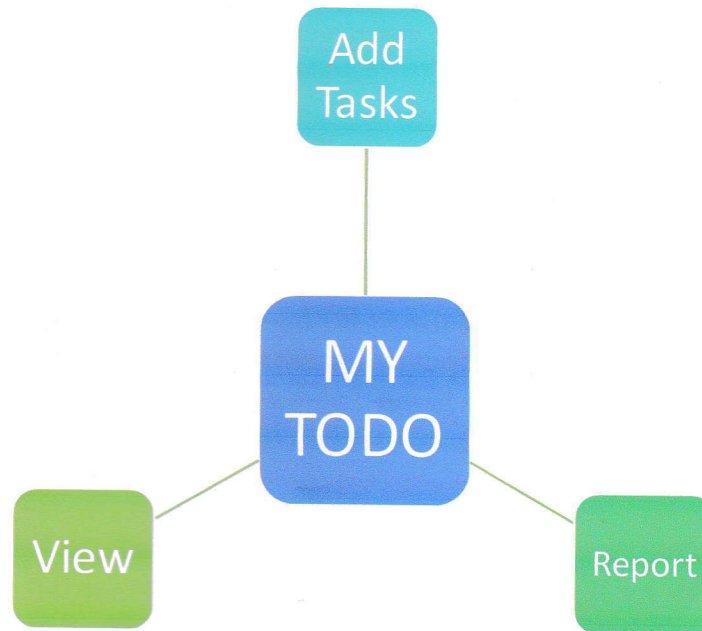


Figure: My ToDo Overview Diagram

My ToDo system carry on three-part

1. Add Tasks Model
2. View Model
3. Report Model

1.4 Descriptions of the project

1.4.1 Product Descriptions

My ToDo is an online-based solution. The objective of this system will help “Daily Tasks” to improve their Tasking skill. This system includes all the system sets needed for the daily tasks. In this system, the features are easily managed and navigate, so users get the report very short time. The user can easily register to the site and complete. My ToDo able to use from any device. User can independently manage edit, delete and completed their daily Tasks.

1.4.2 Project Features

1. **Recorded Tasks:** This one is really simple. One of the advantage of a web based todo list is that you are not limited by ink on paper. Yet, many apps won't let you easily re-organize the order of the todos on your list.
2. **Quick Entry:** My ToDo list should always be ready quickly capture a random task. It shouldn't require more step to enter a todo. They should be quick to allow you to add a todo and get it "out of your head".
3. **Drag and Drop:** In computer graphical user interfaces, drag and drop is a pointing device gesture in which the user selects a virtual object by "grabbing" it and dragging it to a different location or onto another virtual object. In general, it can be used to invoke many kinds of actions, or create various types of associations between two abstract objects.

As a feature, drag-and-drop support is not found in all software, though it is sometimes a fast and easy-to-learn technique. However, it is not always clear to users that an item can be dragged and dropped, or what is the command performed by the drag and drop, which can decrease usability.

The basic sequence involved in drag and drop is:

- Move the pointer to the object
 - Press, and hold down, the button on the mouse or other pointing device, to "grab" the object
 - "Drag" the object to the desired location by moving the pointer to this one
 - "Drop" the object by releasing the button
4. **Simple UI:** Following on from quick capture, overall the system should be simple. Simple to use. Simple user interface. Too many features just clutter things up. And 99% of those extra bells and whistles never get used.
 5. **Today List:** You can't do it all. And one of the best ways to "get work done" is to only concentrate on your top tasks for today.
 6. **View Completed tasks:** Users can easily view your completed tasks.
 7. **Edit and Delete:** ToDo user easily accesses his tasks edit and delete.
 8. **Previous day Tasks:** ToDo system automatically previews uncompleted tasks to the next day.

1.4.3 Project Benefits

I have had many bosses by now. Some have been good and inspiring while some have pretty much made me resolve to look at other paths of revenue rather than a corporate job. However, there was one who was scary yet inspiring and left me with a lot of impressionable moments. I still remember her admonishing me for coming to a meeting without a notebook to jot down notes. One of the times I also learnt from her though was the time that she would march up to me, and demand to see my to-do list just to ensure I had made one. That being my first job, some learnings got imbibed in me for life.

Before my first job, I used to take pride in the fact that I was an impulsive free spirit, or so I liked to imagine myself. An impulsive free spirit could obviously not be bogged down by the shackles of a to-do list. However, after the fore mentioned encounters and through most of my professional life, I have come to recognize just how important and surprisingly liberating a daily to-do list actually is.

1. **BRINGS STRUCTURE TO THE DAY:** There are two ways to go about one's day – either be at sea and just wing it with whatever comes or go about with a list of things to be done. I have realized that my peace of mind is higher when I have a plan rather than swimming around accomplishing anything thrown at my direction. Having a to-do list is like having GPS to reach a destination rather than walking down a road not knowing your destination but just setting on paths shouted out by onlookers.
2. **LESS TO HOLD IN THE HEAD:** Even if we choose not to write down a daily to-do list, mentally we often keep mulling about the tasks that need to be looked into or closed. For me, a to-do list acts almost like the pen sieve in Harry Potter. For the uninitiated, pen sieve is a fabulous device in Albus Dumbledore's office, using which he can extract his memories and anyone can view them as though they are experiencing it. While somewhat different, a daily to-do list helps you extract the tasks floating in your head to find another more concrete place, with less remaining up there.
3. **REDUCES ANXIETY:** Remember Sheldon in Big Bang Theory who has a knack of doing things a certain way and cannot bear the idea of tasks remaining incomplete? Take a look at his obsession for yourself in this video clip. While he might be the epitome of zeigarnik effect, the impact is for real. As discovered by Lithuanian psychologist, Bluma Zeigarnik, waiters remembered even complex orders till the time they were incomplete while failing to recall any detail once an order had been delivered. Probing further, she was able to conclude that our brain focusses much more on incomplete tasks. Incomplete tasks have a way of popping up in our head at unexpected times or to continue brewing in your sub conscious, compounding and leading to anxiety. A daily to-do list helps on cutting down that anxiety.
4. **BETTER TIME MANAGEMENT:** In my experience, keeping daily to-do lists has helped me be better with my time. There was a time when every Monday I would look back and wonder what did I do over the weekend. Gradually, I gave up on my hang up

of not keeping a to-do list for my personal life. I realized if something could be beneficial in my professional life, it could very well help in my personal life too. As I started organizing things in my personal life and keeping a list, be it putting clothes into the washing machine, organizing pins on Tailwind or ordering groceries, I started getting far more done in the same time over a weekend.

5. **BREAKS GOALS INTO ACTION POINTS:** Most of us set ambitious goals for ourselves. These could be over different periods, be it a month, quarter or a year. However, while goals are the result, the tasks that go into channeling the effort towards the goal have a higher probability of getting done when they find their way on a to-do list.

1.5 Design and Implementation

Our web application is being developed by HTML, CSS, jQuery, JavaScript, PHP and MySQL, we have used Xampp server to run our application. We designed the following

- Frontend (use for all)
- Authentication (for user, to register, login and managing accounts info)

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer. With the advantage a number of common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

CHAPTER - 2
OBJECT ORIENTED DESIGN

2.1 Introduction of Object-Oriented Design

Object-Oriented Design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design.

An object contains encapsulated data and procedures grouped together to present an entity. The 'object interface', how the object can be interacted with, is also defined. An object-oriented program is described by the interaction of these objects. Object-Oriented design is the discipline of defining the objects and their interactions to solve a problem that was identified and documented during object-oriented analysis.

Some typical input artifacts for object-oriented design are:

- Use Case Diagram
- Activity Diagrams
- Class Diagram
- Entity Relationship Diagram

2.2 Use Case Diagram

In software and system engineering a use case is a case in the use of a system is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

2.2.1 Use case of user Registration

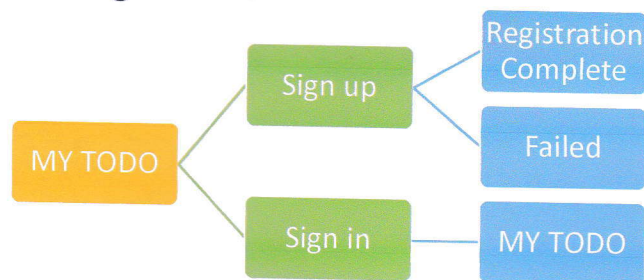


Figure: User Registration

As a user respondent open the website then register their account, and finally complete their registration then login into My ToDo.

2.2.2 Use case of user

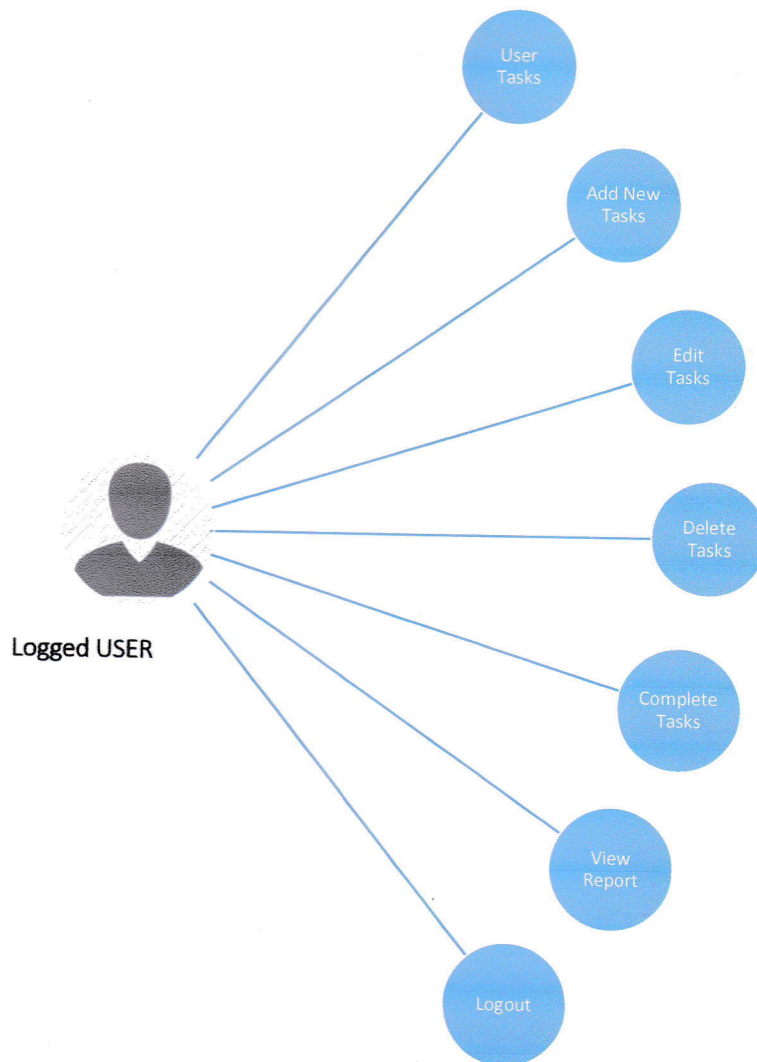


Figure: User Use Case

User is the most important person. Valid user login into My ToDo and they can use usually My ToDo features. It's very simple for user, no hard task in this system. Users can easily view his task, add new tasks, edit tasks, delete tasks, view complete tasks and finally view report as his requirements.

2.3 Activity Diagram

Activity diagram means to describe sequencing and conditions of actions. Such descriptions commonly are work flow, control flow and object flow models.

Simple Activity Diagrams consist of:

- Initial node
- Activity final node
- Activity between

2.3.1 Activity diagram of User Registration

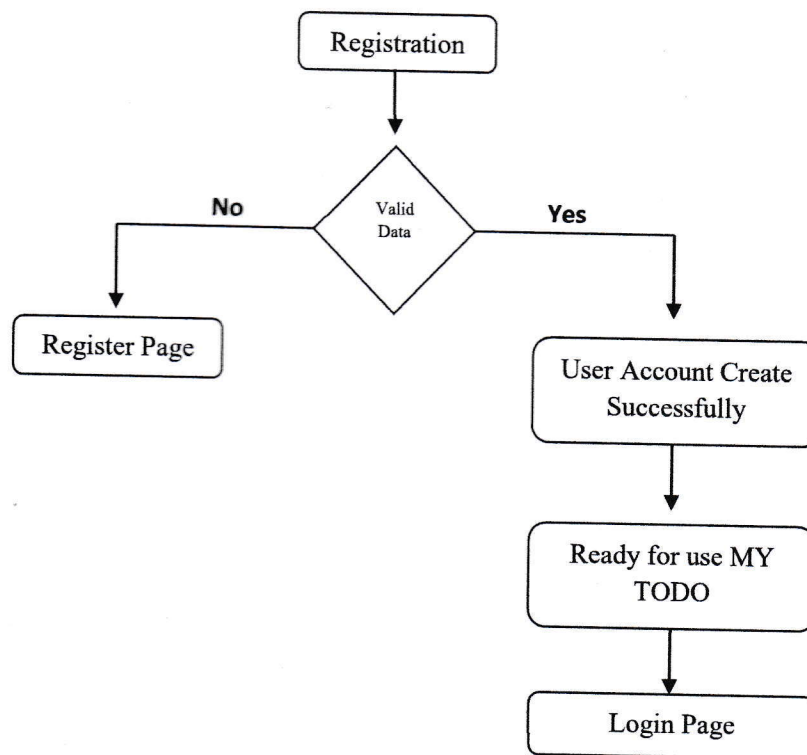


Figure: Registration Activity Diagram

Registration activity diagram shows the entire process of user registration. A registered user is one who uses a program or a website and provides his/her credentials, effectively proving his/her identity. Generally speaking, any person can become a registered user by providing some credentials, usually in the form of a username (or email) and password. After that, one can access information and privileges unavailable to non-registered users. The action of providing the proper credentials for a website is called logging in, or signing in.

2.3.2 Activity diagram of User

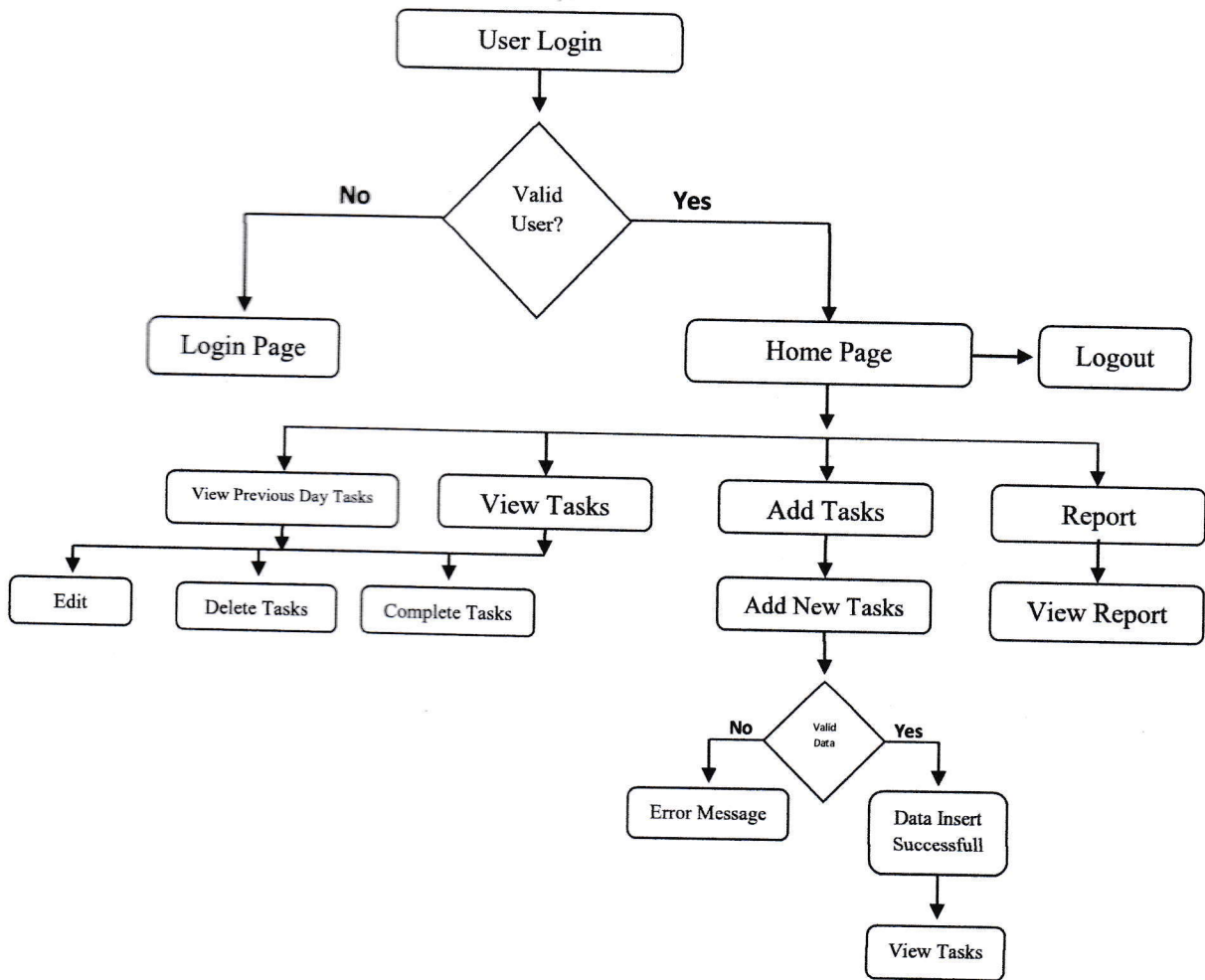


Figure: User Activity Diagram

User activity diagram shows the entire activities of a register user. Firstly, the user can view our site then the user easily creates his account. If already create his account he/she is ready to access our daily task management system. A valid user always permitted to access our system.

The user can permission to log in to his valid user name and valid password. When a user inputs his username and password submitted to our system, our program checked his data and cross-match his data with the database, then the program automatically decided his data is wrong or right if the program found right data then he can log in into this system.

2.4 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

2.4.1: Class Diagram of User

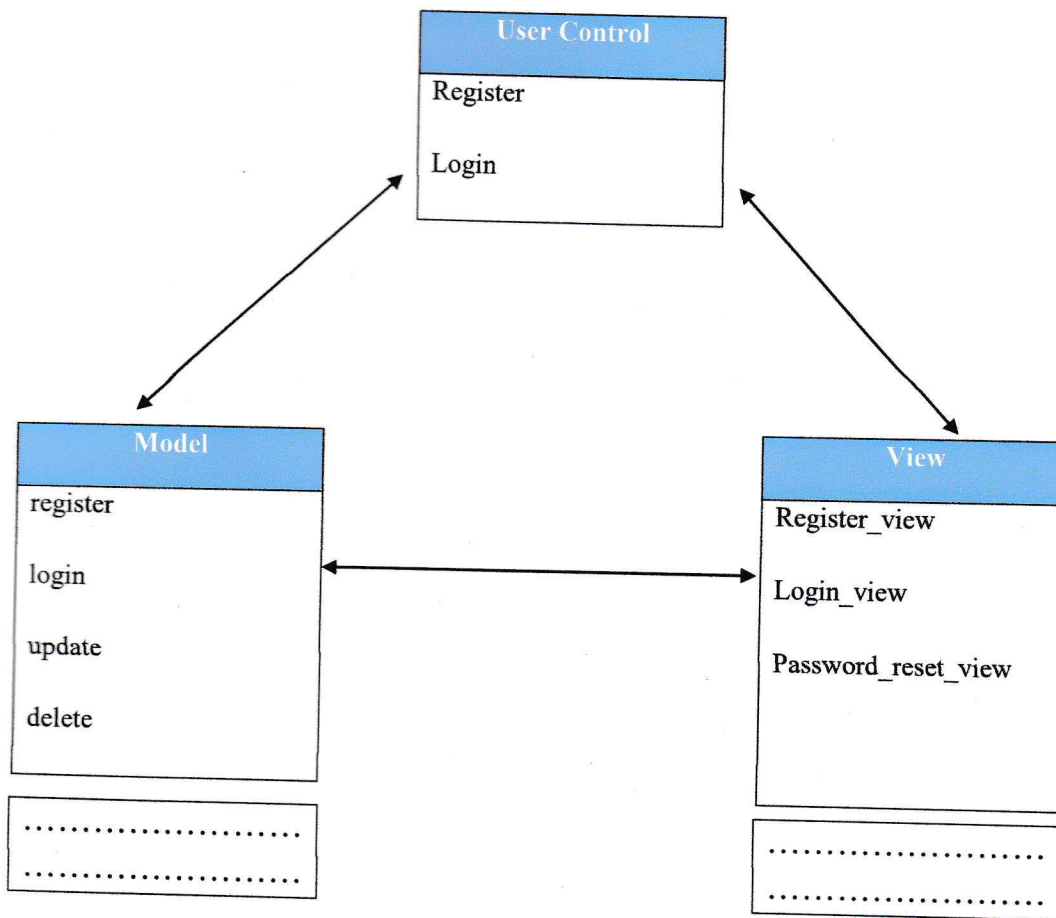


Figure: Class Diagram of User Control

2.4.1: Class Diagram of Add Tasks

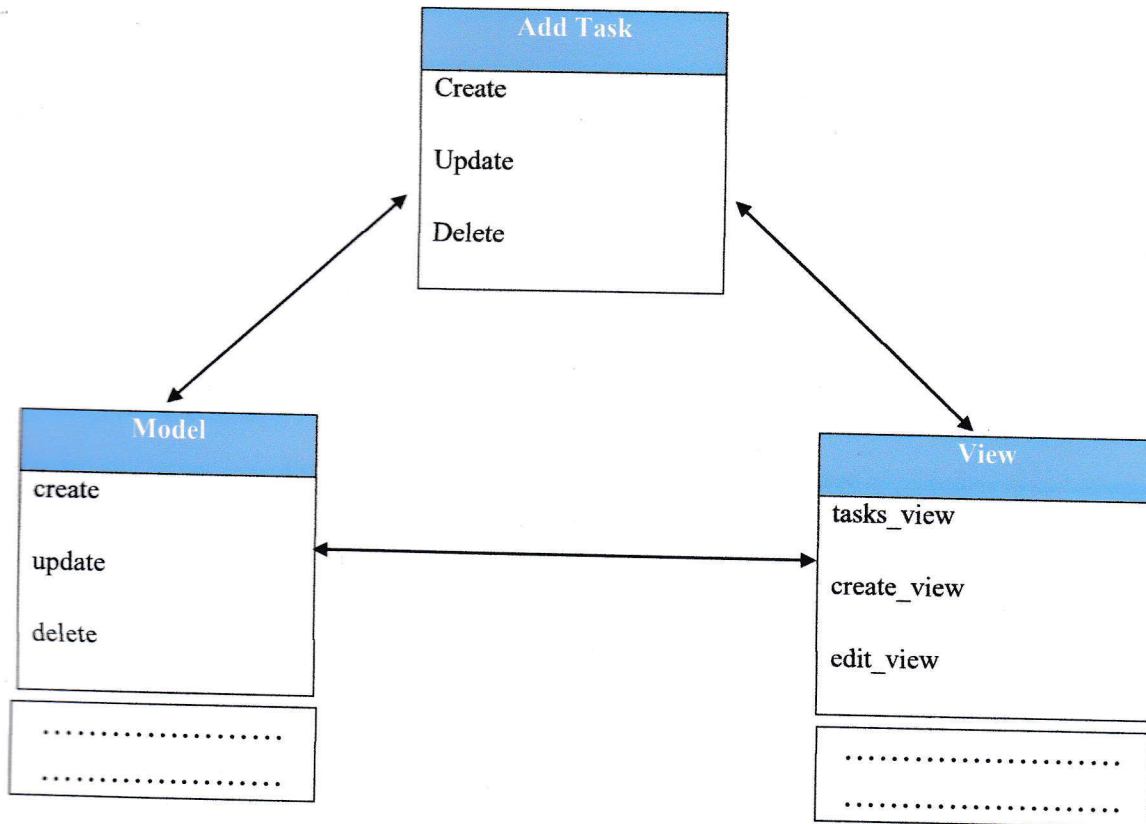


Figure: Class Diagram of add tasks

2.5 Entity Relationship Diagram

In software engineering, an Entity – Relationship model (ER model for short) is an abstract way to describe a database. It usually starts with a relational database, which stores data in tables. Some of the data in these tables point to data in other tables – for instance, your entity in the database could point to several entries for each of the phone numbers that are yours. The ER model would say that you are an entity, and each phone number is an entity, and the relationship between you and the phone numbers is ‘has a phone number’. Diagrams created to design these entities and relationships are called entity-relationship diagrams or ER diagrams. There are four Types of ER Diagram in our system:

2.5.1 ER Diagram of our system:

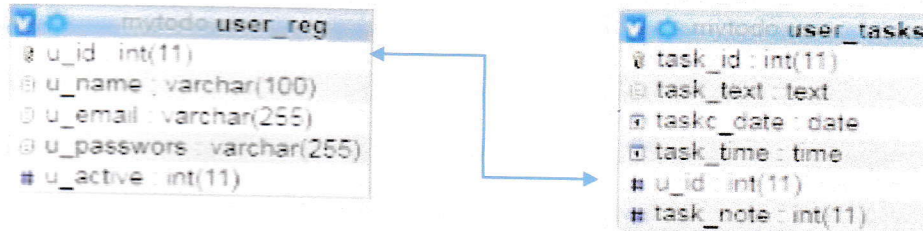


Figure: ER Diagram of MY TODO Database

There are two tables in our database.

1. User Registrations
2. User tasks

User Registration: The user registration table collects user registration information and user active status.

User Tasks: The user tasks table collects all user tasks information.

For example,

1. Tasks Create date and times.
2. User ID.
3. Tasks completed or uncompleted.
4. Tasks Text.

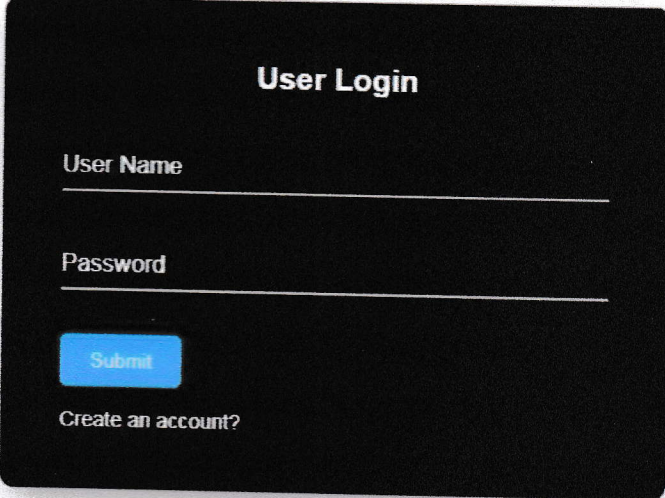
CHAPTER - 3
USER INTERFACE DESIGN

3.1 Introduction of User Interface Design

The user interface, in the industrial design field of human-machine interaction, is the space where interaction between humans and machines occurs. The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions. Examples of this broad concept of user interfaces include the interactive aspects of computer operating system, hand tools, heavy machinery operator controls and process controls. A user interface is the system by which people (users) interact with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of:

- Input, allowing the users to manipulate a system
- Output, allowing the system to indicate the effects of the user's manipulation

3.1.1 User Login UI

A screenshot of a user login interface. The title "User Login" is centered at the top. Below it are two input fields: "User Name" and "Password", each with a horizontal line for text entry. A blue "Submit" button is positioned below the password field. At the bottom left, there is a link that says "Create an account?".

User Login

User Name _____

Password _____

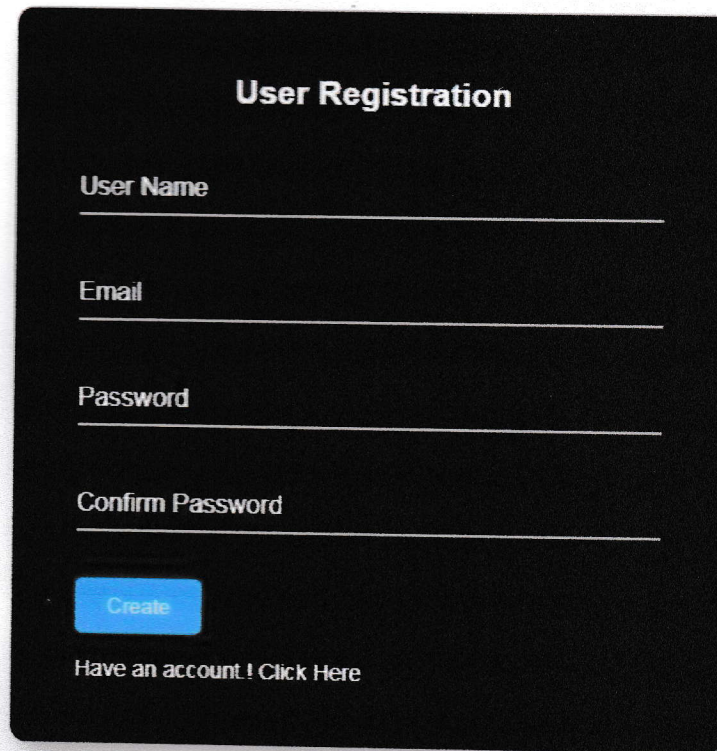
[Submit](#)

[Create an account?](#)

Figure: Login Page

This is the home page and user login page of our project. Thought this page register used can log in My ToDo web application by providing correct username and password. Fix making registering a new user need to click on "Create an account?" link.

3.1.2 User Registration UI



The image shows a dark-themed user registration form. At the top, the title "User Registration" is centered in white. Below the title are four input fields, each with a white label and a white underline: "User Name", "Email", "Password", and "Confirm Password". At the bottom left of the form is a blue button with the text "Create" in white. Below the button is a link that says "Have an account! Click Here" in white text.

Figure: User Registration Page

The user registration page is the most important for user registration. For accessing our “My ToDo” web application a user must fill-up the user registration form by providing appropriate information. Without registration, no one can access our software.

3.1.3 Add Task UI

A black-bordered modal window titled "Add Task". It contains a white text input field with the placeholder text "Enter Text". Below the input field are two buttons: a green button labeled "Save" and a red button labeled "Close".

Figure: Add Task

Add tasks is to create your todo list you can write your text or tasks here and click the save button.

3.1.4 Edit Task UI

A black-bordered modal window titled "Edit Task". It contains a white text input field with the placeholder text "Test Task". Below the input field are two buttons: a green button labeled "Update" and a red button labeled "Close".

Figure: Edit Task

The user can easily edit his tasks from this form. the user just clicks edit button then edit his task and click the update button.

3.1.5 ToDo List UI

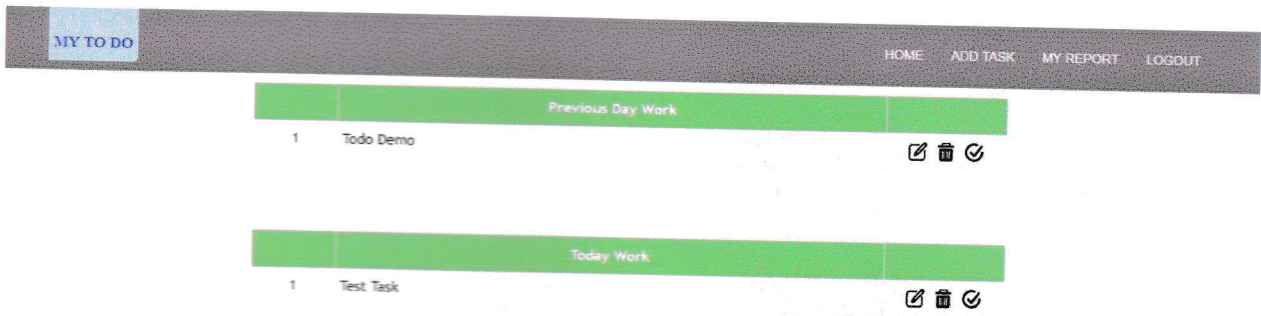


Figure: ToDo List View

This is the homepage of the todo list. This page user can view his todo, edit his todo, completed his todo and drag and drop his todo as his priority also user can view previous days uncompleted lists.

3.1.6 Report UI

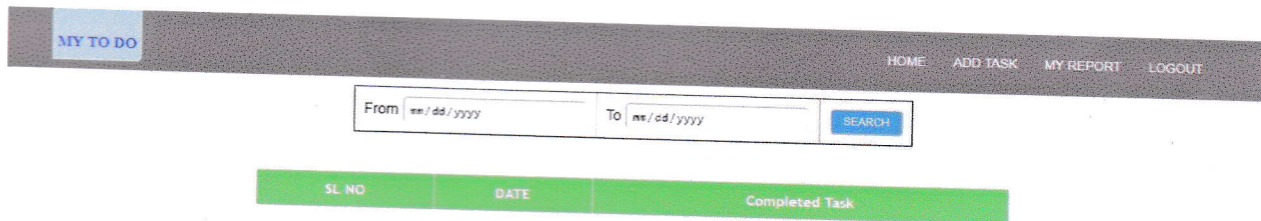


Figure: Report View Page

The user can view his completed tasks. Firstly, click **MY REPORT** page then select from date - to date then click the search button. finally, completed tasks view on this page.

3.1.7 Report UI

MY TO DO

HOME ADD TASK MY REPORT LOGOUT

From To

SL. NO	DATE	Completed Task
1	2019-09-03	<ul style="list-style-type: none">• Test Task• Todo Demo

Figure: Report View 1

CHAPTER – 4
CONCLUSION AND FUTURE
WORK

4.1 Conclusion:

"Online Daily Task Management System" is an efficient task management system to gather your thoughts and ideas in one place. You can create tasks, make to-do lists, and make tasks do over so that you don't have to remember everything.

4.2 Future Work:

- ✓ Create sub-tasks, and add comments
- ✓ Make to-do lists with notes
- ✓ Email notifications and SMS reminders
- ✓ Add voice record with your daily tasks
- ✓ Add reminders and events in Calendar
- ✓ Share your lists and work collaboratively on projects with your colleagues, friends, and family

References

1. Francis J. Parker, "A perfect guide to stress free productivity, task management and creating to-do lists", Rescue Your To-Do List, 2018.
2. Darlene Tucker, "Checklist and Self Discipline for Personal Success", Task Management, 2018.
3. S.J. Scott, "A Simple Guide to Getting the Important Things Done", To-Do List Makeover, 2014.
4. Vanessa Robins, "List of Thing to do and Journal task easy and simple time management", To Do List Notepads, 2018.
5. Rogue Plus Publishing, "Checklist Blank, To Do And Notes, Daily Task Sheets, To Do List Notebook Kids, Agenda Notepad For Men, Women, Students & Kids, Hydrangea Flower Cover", To Do List Notebook, 2018.
6. Damon Zahariades, "A Stress-Free Guide to Creating To-Do Lists That Work! (Kindle Edition)", To-Do List Formula, 2016.
7. Michael Linenberger, "Quickly Get Your Chaos Completely Under Control Kindle Edition", The One Minute To-Do List, 2011.
8. Andrew Kunesh, "8 Powerful Ways to Manage Your Tasks", Your To Do List and Beyond, Published March 11, 2019.
9. Elementum Money. Published on: February 2, 2019 [Online] Available at: <https://elementummoney.com/to-do-list-benefits/>. [Accessed 08 August 2019]
10. Wikipedia. 2019. Apache Friends. Published on: May 22, 2002 [Online] Available at: <https://en.wikipedia.org/wiki/XAMPP>. [Accessed 04 August 2019]

APPENDIX

index.php

```
<?php session_start(); ?>
<?php
$link = mysqli_connect('localhost', 'root', '', 'number') or die("<script
language='javascript'>alert('Unable to connect to database')</script>");

if (!empty($_POST)) {
    $name = $_POST['name'];
    $password = $_POST['password'];
    $query = "SELECT * FROM users WHERE username= '$name'";
    $res = mysqli_query($link, $query);
    $data = mysqli_fetch_assoc($res);
    if ($data['password'] == md5($password)) {
        $_SESSION['uid'] = $data['u_id'];
        $_SESSION['uemail'] = $data['u_email'];
        $_SESSION['time'] = time();

        header('location:home.php');
    } else {
        echo "Login Failed";
    }
}
?>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="style/style.css" type="text/css">
<title>Login Page</title>
</head>
<body>
<div class="box">
<h2> User Login</h2>
<form action="" method="post">
<div class="inputbox">
<input type="text" name="name" required="">
<label>User Name</label>
</div>
<div class="inputbox">
<input type="password" name="password" required="">
<label>Password</label>
</div>
<input type="submit" name="submit" value="Submit">
</form>
<a href="userreg.php" style="color:white; font-size: 14px; text-decoration: none;
padding-top: 20px;">Create an account?</a>
</div>
</body>
</html>
```

userreg.php

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'number') or die("<script
language='javascript'>alert('Unable to connect to database')</script>");
//print_r($_POST);
if (!empty($_POST)) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $cpass = $_POST['cpass'];
    $email = $_POST['email'];
    $query = mysqli_query($link, "SELECT * FROM `users` WHERE username ='" .
$username . "' or email ='" . $email . "'");
    if (!$query) {
        die("Error: " . mysqli_error($link));
    }
    if (mysqli_num_rows($query) > 0) {
        echo "<script language='javascript'>alert('User Name or Email Already
exists!')</script>";
        //echo "Name Already exists";
    }
    //for password
    elseif ($password != $cpass) {
        echo "Confirm Password not match";
    }
    //for user name
    elseif (!preg_match("/^[a-zA-Z]*$/", $username)) {
        $nameErr = "User Name Only letters allowed";
        echo $nameErr;
    } elseif (!preg_match("#[0-9]+#", $password)) {
        $passwordErr = "Your Password Must Contain At Least 1 Number!";
        echo $passwordErr;
    } elseif (!preg_match("#[A-Z]+#", $password)) {
        $passwordErr = "Your Password Must Contain At Least 1 Capital Letter!";
        echo $passwordErr;
    } elseif (!preg_match("#[a-z]+#", $password)) {
        $passwordErr = "Your Password Must Contain At Least 1 Lowercase Letter!";
        echo $passwordErr;
    } else {
        if (!empty($email)) {
            if (preg_match("/^[_0-9a-zA-Z-]+@[0-9a-zA-Z][0-9a-zA-Z-]+\.[a-zA-
Z]{2,6}$/i", $email)) {
                // Return Error - Invalid Email
                $query = "insert into `users` (`username`, `email`, `password`) VALUES
($username, '$email', md5('$password'))";
                $res = mysqli_query($link, $query);
                mysqli_close($link);
                if ($res == TRUE) {
                    echo "<script language='javascript'>alert('Save Successfully')</script>";
                } else {
```



```

        echo"<script language='javascript'>alert('User Name Aready Use')</script>";
    }
} else {

    $msg = 'The email you have entered is invalid, please try again.';
    echo $msg;
}
}
}
}
?>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="style/style.css" type="text/css">
<title>Login Page</title>
</head>
<body>
<div class="box">
<h2> User Registration</h2>
<form action="userreg.php" method="POST">
<div class="inputbox">
<input type="text" name="username" required="">
<label>User Name</label>
</div>
<div class="inputbox">
<input type="email" name="email" required="">
<label>Email</label>
</div>
<div class="inputbox">
<input type="password" name="password" required="">
<label>Password</label>
</div>
<div class="inputbox">
<input type="password" name="cpassword" required="">
<label>Confirm Password</label>
</div>
<input type="submit" name="submit" value="Create">
</form>
<a href="index.php" style="color:white; font-size: 14px; text-decoration: none;
padding-top: 20px;">Have an account.! Click Here</a>
</div>
</body>
</html>

```

home.php

```
<?php require_once './resource/header.php'; ?>
<?php
if (empty($_SESSION['uid'])) {
    header('location:index.php');
}
date_default_timezone_set('asia/dhaka');
$date = date('Y-m-d');
$prev_date = date('Y-m-d', strtotime($date . '-1 day'));
$uid = $_SESSION['uid'];
$link = mysqli_connect('localhost', 'root', '', 'number');
$query = "select * from `todo` where tddone='0' and td_date='$date' and u_id='$uid' order by
position_order";
$queryer = "select * from `todo` where tddone='0' and td_date='$prev_date' and u_id='$uid'
order by td_id desc ";
$result = mysqli_query($link, $query);
$result = mysqli_query($link, $queryer);
?>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-
ui.min.js"></script>
<table id="customers" align="center">
    <tr>
        <th></th>
        <th><?php
            $query = "select td_date from `todo` where td_date='$prev_date' and tddone='0' and
u_id='$uid' group by td_date";
            $result = mysqli_query($link, $query);
            if (!empty($result)) {
                while ($row = mysqli_fetch_assoc($result)) {
                    if ($prev_date) {
                        echo 'Previous Day TODO';
                    }
                }
            }
        ?></th>
        <th></th>
    </tr>
    <?PHP if (empty($result)) { ?>
        <tr>
            <th>SL NO</th>
            <th>Task</th>
            <th>Action</th>
        </tr>
        <?PHP } ?>
        <?php {
            $sino = 1;
            if (!empty($result)) {
                while ($data = mysqli_fetch_assoc($result)) {
```

```

?>
<tr>
  <td align="center" width="80px"><?php echo $slno++; ?></td>
  <td><?php echo $data['td_text']; ?></td>
  <td class="" align="center" width="120px">
    <button title="Edit!" style="padding: 2px; cursor:pointer; background-
color:transparent; border-color:transparent;" onclick="location.href = 'edit.php?id=<?php
echo $data['td_id'] ?>' type="button"></button>
    <button title="Delete!" style="padding: 2px; cursor:pointer; background-
color:transparent; border-color:transparent;" onclick="location.href =
'tddelete.php?did=<?php echo $data['td_id'] ?>' type="button"> </button>
    <button title="Complete!" style=" padding: 2px; cursor:pointer; background-
color:transparent; border-color:transparent;" onclick="ConfirmDone0()" type="button"></button>
  </td>
  <script type="text/javascript">
    function ConfirmDone0()
    {
      var baseUrl = 'tddone.php?id=<?php echo $data['td_id'] ?>';
      if (confirm("Are You Sure.! Complete your task?"))
        location.href = baseUrl;
    }
  </script>
</tr>
<?php
}
}
}
}
?>
</table>
<table id="customers" align="center" >
  <tr>
    <th></th>
    <th><?php
      $sque = "select td_date from `todo` where tddone='0' and u_id='$uid' group by
td_date";
      $res = mysqli_query($link, $sque);

      if (!empty($res)) {
        while ($row = mysqli_fetch_assoc($res)) {
          if ($row['td_date'] == $date) {
            echo 'Todays TODO';
          } else {
          }
        }
      }
    ?></th>
    <th></th>

```



```

</tr>
<?PHP if (empty($res)) { ?>
  <tr>
    <th>SL NO</th>
    <th>Task</th>
    <th>Action</th>
  </tr>
<?PHP } ?>
<tbody class="row_position">
  <?php
  require('db_config.php');
  {
    $sl = 1;
    if (!empty($result)) {
      while ($row = mysqli_fetch_assoc($result)) {
        ?>
        <tr id="<?php echo $row['td_id'] ?>">
          <td align="center" width="80px"><?php echo $sl++; ?></td>
          <td><?php echo $row['td_text']; ?></td>
          <td class="" align="center" width="120px">
            <button title="Edit!" style="padding: 2px; cursor:pointer; background-
            color:transparent; border-color:transparent;" onclick="location.href = 'edit.php?id=<?php
            echo $row['td_id'] ?>' type="button">
            </button>
            <button title="Delete!" style="padding: 2px; cursor:pointer; background-
            color:transparent; border-color:transparent;" onclick="location.href =
            'tddelete.php?did=<?php echo $row['td_id'] ?>' type="button">
            </button>
            <button title="Complete!" style="padding: 2px; cursor:pointer;
            background-color:transparent; border-color:transparent;" onclick="ConfirmDone()"
            type="button">
            </button>
          </td>
          <script type="text/javascript">
            function ConfirmDone()
            {
              var baseUrl = 'tddone.php?id=<?php echo $row['td_id'] ?>';
              if (confirm("Are You Sure.! Complete your task?"))
                location.href = baseUrl;
            }
          </script>
        </tr>
      <?php
    }
  }
  ?>
</tbody>

```



```

<script type="text/javascript">
    $(".row_position").sortable({
        delay: 150,
        stop: function () {
            var selectedData = new Array();
            $('.row_position>tr').each(function () {
                selectedData.push($(this).attr("id"));
            });
            updateOrder(selectedData);
        }
    });

function updateOrder(data) {
    $.ajax({
        url: "ajaxPro.php",
        type: 'post',
        data: {position: data},
        success: function () {

            alert('your change successfully saved');

        }

    })

}
</script>
</table>
</body>
<?php require_once './resource/footer.php'; ?>
</html>

```

insert.php

```

<?php require_once './resource/header.php';
$link = mysqli_connect('localhost', 'root', '', 'number');
if (empty($_SESSION['uid'])) {
    header('location:index.php');
}
//print_r($_POST);
date_default_timezone_set('asia/dhaka');
if (!empty($_POST)) {
    $a_name = $_POST['title'];
    $date = date('Y-m-d');
    $time = date('H:i:s');
    $uid = $_SESSION['uid'];
    if (!empty($a_name)) {
        if (!preg_match("/[A-Za-z0-9]+/", $a_name)) {

```

```

        $nameErr = "Only Text And Number";
    } else {
        $query = "INSERT INTO `todo`(`td_text`, `td_date`, `td_time`, `u_id`) VALUES(
'$a_name','$date','$time','$uid')";
        $res = mysqli_query($link, $query);
        if ($res) {
            echo "Add Successful";
            header('location:home.php');
        } else {
            echo "Add Failed";
        }
    }
}
}
?>
<div>
    <div class="form-popup">
        <h2 style="color: white;">Add Task</h2>
        <?php if (!empty($nameErr)) echo $nameErr; ?>
        <form action="insert.php" class="form-container" method="post">
            <input type="text" placeholder="Enter Text" name="title" required>
            <button type="submit" class="btn">Save</button>
            <button type="button" class="btn cancel" onclick="location.href =
'home.php'">Close</button>
        </form>
    </div>
</div>
<?php require_once './resource/footer.php'; ?>

```

edit.php

```

<?php
require_once './resource/header.php';
if (empty($_SESSION['uid'])) {
    header('location:index.php');
}
if (!empty($_GET)) {
    $eid = $_GET['id'];
    $link = mysqli_connect('localhost', 'root', '', 'number');
    $qry = "select * from todo where td_id='$eid'";
    $result = mysqli_query($link, $qry);
    $data = mysqli_fetch_assoc($result);
}

if (!empty($_POST)) {
    $a_name = $_POST['name'];
    if (!empty($a_name)) {
        $query = "update todo set td_text='$a_name' where td_id='$eid'";
        $res = mysqli_query($link, $query);
    }
}

```

```

    if ($res) {
        header('location:home.php');
    } else {
        echo"Update Failed";
    }
}
//header('location:index.php');
}
?>

```

```

<form method="POST" action="" class="f-container" >
  <div align="center" style="margin-top: 40px; border:10px; border-radius:10px; position:
absolute;
  top: 50%;
  left: 50%;
  padding: 40px;
  background: rgba(0,0,0,.8);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0,0,0,.5);
  transform: translate(-50%,-50%);">
    <h2 style="color: white;">Edit Task</h2>
    <input type="text" name="name" required value="<?php echo $data['td_text']; ?>">
    <button type="submit" class="btn">Update</button>
    <button type="button" class="btn cancel" onclick="location.href =
'home.php'">Close</button>

  </div>
</form>

```

tddone.php

```

<?php
$link = mysqli_connect('localhost', 'root', '', 'number');
if (!empty($_GET)) {
    $id = $_GET['id'];
    $query = "update todo set tddone='1' where td_id='$id'";
    $result = mysqli_query($link, $query);
    if($result){
        header('location:home.php');
    }
}
?>

```

tddelete.php

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'number');
if (!empty($_GET)) {
    $did = $_GET['did'];
    $query = "delete from todo where td_id='$did'";
    $result = mysqli_query($link, $query);
    if($result){
        header('location:home.php');
    }
}
?>
```

report.php

```
<?php
require_once './resource/header.php';
?>
<?php
$link = mysqli_connect('localhost', 'root', '', 'number');
if (empty($_SESSION['uid'])) {
    header('location:index.php');
}
if (isset($_GET["date"])) {
    $uid = $_SESSION['uid'];
    $valueToSearch = $_GET['date'];
    $valueToSearch1 = $_GET['date1'];
    $query = "select * from `todo` where tddone='1' AND u_id='$uid' AND td_date
    BETWEEN '$valueToSearch' AND '$valueToSearch1' GROUP BY td_date";
    $result = mysqli_query($link, $query);
} else {
}
?>
<form action="report.php" method="GET">
    <table border="1px" style="border-collapse: collapse; margin-top: 80px;"
    cellpadding="10" align="center">
        <tr>
            <td> From <input class="textb" type="date" name="date" required="" > </td>
            <td> To <input class="textb" type="date" name="date1" required="" > </td>
            <td><input class="addBtn" type="submit" name="search" value="SEARCH"></td>
        </tr>
    </table>
    <br>
    <table id="customers" align="center" style="text-align: center;" >
        <tr>
            <th bgcolor="#FF8066">SL NO</th>
```



```

        <th bgcolor="#FF8066">DATE</th>
        <th bgcolor="#FF8066">Completed Task</th>
    </tr>
    <?php
    $sl = 1;
    if (!empty($result)) {
        while ($row = mysqli_fetch_assoc($result)) {
            ?>
            <tr>
                <td><?php echo $sl++; ?></td>
                <td align="center"><?php echo $row['td_date'];
    $dt = $row['td_date'];
            ?>
                </td >
                <td align="left">
                    <?php
                    $sque = "select * from `todo` where tddone='1' AND u_id='$uid' and td_date =
'Sdt' group by td_id";
                    $res = mysqli_query($link, $sque);
                    if (!empty($res)) {
                        while ($row = mysqli_fetch_assoc($res)) {
                            echo'<li>' . $row['td_text'] . '</li>';
                        }
                    } else {
                    }
                    ?>
                </td>
            <?php
        }
    }
    ?>
</tr>
</table>

```

```
<?php require_once './resource/footer.php';?>
```

logout.php

```

<?php
session_start();
session_destroy();
header('location:index.php');
?>

```

header.php

```

<?php session_start(); ?>
<html>
<head>

```

```

<meta charset="UTF-8">
<link rel="stylesheet" href="style/style.css" type="text/css" >
<title>My TODO</title>
</head>
<body>
  <nav>
    <div class="brand">
      <h2>MY TO DO</h2>
    </div>
    <ul>
      <li><a href="home.php">HOME</a></li>
      <li><a href="insert.php">ADD TASK</a></li>
      <li><a href="report.php">MY REPORT</a></li>
      <li><a href="logout.php">Logout</a></li>
    </ul>
  </nav>
</body>
</html>

```

footer.php

```

</div>
<style>
  .footer {
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: rgba(0,0,0,0.6);
    color: white;
    text-align: center;
  }
</style>
<div class="footer">
  <p><footer>&copy; Copyright 2019 My TODO</footer>
</p>
</div>
</body>
</html>

```

ajaxpro.php

```

<?php
require('db_config.php');
$position = $_POST['position'];
$i = 1;
foreach ($position as $k => $v) {
  $sql = "Update todo SET position_order=" . $i . " WHERE td_id=" . $v;
  $mysqli->query($sql);
  $i++;
}

```

```
    header('location:home.php');
}
?>
```

db_config.php

```
<?php
$db = new mysqli("localhost", "root", "", "number");
?>
```

style.css

```
body{
    margin: 0;
    padding: 0;
    font-family: sans-serif;
    background: #fff;
    background-size: cover;
    font-size: 14px;
}
table tr td .textb{
    padding: 5px;
    width: 200px;
    height: 30px;
    border-radius: 5px;
}
table tr td .addBtn
{
    padding: 5px;
    width: 80px;
    height: 30px;
    background: #2471A3;
    color: #fff;
    text-align: center;
    font-size: 12px;
    cursor: pointer;
    transition: 0.3s;
    border-radius: 5px;
}
.addBtn:hover {
    background-color: #27AE60;
}
.box{
    position: absolute;
    top: 50%;
```

```

left: 50%;
transform: translate(-50%,-50%);
width: 450px;
padding: 40px;
background: rgba(0,0,0,.8);
box-sizing: border-box;
box-shadow: 0 15px 25px rgba(0,0,0,.5);
border-radius: 10px;
}
.box h2{
margin: 0 0 30px;
padding: 0;
color: #fff;
text-align: center;
}

.box .inputbox {
position: relative;
}
.box .inputbox input{

width: 100%;
padding: 10px 0;
font-size: 14px;
color: #fff;
margin-bottom: 30px;
border: none;
border-bottom: 1px solid #fff;
outline: none;
background: transparent;
}

.box .inputbox label{
position: absolute;
top: 0;
left: 0;
padding: 10px 0;
font-size: 16px;
color: #fff;
pointer-events: none;
transition: .5s;
}
.box .inputbox input:focus ~ label,
.box .inputbox input:valid ~ label
{
top:-18px;
left: 0;
color: #03a9f4;
font-size: 12px;
}

```



```
}
```

```
.box input[type="submit"]{  
  background: transparent;  
  border: none;  
  outline: none;  
  color:#fff;  
  background: #03a9f4;  
  padding: 10px 20px;  
  cursor: pointer;  
  border-radius: 5px;
```

```
}
```

```
.mar{  
  margin-top: 100px;  
  height: 290px;
```

```
}
```

```
.mar .one{  
  padding: 50px;
```

```
}
```

```
nav{  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 70px;  
  background: rgba(0,0,0,0.6);  
  padding:0 50px;  
  box-sizing: border-box;  
  font-family: sans-serif;
```

```
}
```

```
nav .brand  
{  
  float: left;  
  height: 100%;  
  line-height: 70px;
```

```
}
```

```
nav .brand h2  
{  
  margin: 0;  
  padding: 5px;  
  color: blue;  
  font-size: 17px;  
  background-color:lightblue;
```

```

height: 50px;
border-radius: 6px 0;
font-family: Times;
}
nav ul {
float: right;
display: flex;
margin: 0;
padding: 0;
}
nav ul li {
list-style: none;
}
nav ul li a {
position: relative;
display: block;
height: 70px;
line-height: 70px;
padding: 0 15px;
box-sizing: border-box;
color: #fff;
text-decoration: none;
text-transform: uppercase;
transition: .5s;
}
nav ul li a: hover {
color: #262626;
}
nav ul li a: before
{
content: "";
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
background: #fff;
transform-origin: right;
z-index: -1;
transform: scaleX(0);
transition: transform .5s;
}
nav ul li a: hover: before
{
transform-origin: left;
transform: scaleX(1);
}

```

```
.one tr td input{  
  border-radius: 4px;  
  width: 200px;  
  height: 30px;  
  margin: 1px;  
  box-sizing: border-box;  
  padding: 8px;  
  font-family: sans-serif;  
  font-size: 14px;
```

```
}
```

```
.one tr td .btn {  
  padding:0;  
  font-size: 15px;  
  color: white;  
  background: #FF6F91;  
  border: none;  
  cursor: pointer;  
  border-radius: 5px;
```

```
}
```

```
.vtab tr th  
{  
  font-size: 10px;  
  padding: 5px;  
  color: #fff;  
  background-color: lightslategray;
```

```
}
```

```
.view{  
  height: 300px;  
  font-size: 14px;  
  padding-left: 50px;  
  padding-right: 50px;
```

```
}
```

```
.vtab tr .sl  
{  
  width: 50px;
```

```
}
.vtab tr .s2
{
  width: 200px;

}
.vtab tr .s3
{
  width: 150px;
  text-align: center;

}
.vtab tr .s4
{
  width: 150px;

}
.vtab tr .s5
{
  width: 600px;

}

.view .ser tr td{
  padding: 5px;
  width: 80%;

}

.view .ser tr td .button{
  padding: 5px;
  width: 80px;
  height: 30px;
  background: #2471A3;
  color: #fff;
  text-align: center;
  font-size: 12px;
  cursor: pointer;
  transition: 0.3s;
  border-radius: 5px;
}

.view .ser tr td .button:hover {
  background-color: #27AE60;
}

.view .ser tr td .rabbi{
  margin: 0;
  padding: 5px;
```



```

width: 200px;
height: 30px;
border-radius: 5px;
}

.view .vtab tr td .btn {
padding: 8px;
font-size: 15px;
color: white;
background: #48C9B0;
border: none;
cursor: pointer;
border-radius: 5px;
text-decoration: none;
}
.view .vtab tr td .btn1 {
padding: 8px;
font-size: 15px;
color: white;
background: #FF6F91;
border: none;
cursor: pointer;
border-radius: 5px;
text-decoration: none;
margin: 10px;
}
.view .vtab tr .tdbtn {
text-align: center;
padding: 15px;
}
.view .vtab .VIEW{
font-family: sans-serif;
font-size: 14px;
height: 15px;
white-space: normal;
padding: 2px;
}

```

homepage style css

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-
ui.min.js"></script>
<style>
#customers {

```

```

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
border-collapse: collapse;
width: 60%;
background: #ddd;
margin-top: 80px;

}

#customers td, #customers th {
border: 1px solid #ddd;
padding: 5px;

}

#customers tr:nth-child(even){background-color: #fdfdfd;}

#customers tr:hover {background-color: #fffddd;}

#customers th {
padding-top: 10px;
padding-bottom: 10px;
text-align: center;
background-color: #4CAF50;
color: white;
}
</style>

```

insert page style.css

```

.butn {
-webkit-border-radius: 8;
-moz-border-radius: 8;
border-radius: 8px;
-webkit-box-shadow: 4px 5px 5px #e0e0e0;
-moz-box-shadow: 4px 5px 5px #e0e0e0;
box-shadow: 4px 5px 5px #e0e0e0;
font-family: Georgia;
color: #ffffff;
font-size: 14px;
padding: 13px;
background: #fa8a00;
text-decoration: none;
}

.butn:hover {
background: #3cb0fd;
background-image: -webkit-linear-gradient(top, #3cb0fd, #3498db);
background-image: -moz-linear-gradient(top, #3cb0fd, #3498db);
}

```

```

background-image: -ms-linear-gradient(top, #3cb0fd, #3498db);
background-image: -o-linear-gradient(top, #3cb0fd, #3498db);
background-image: linear-gradient(to bottom, #3cb0fd, #3498db);
text-decoration: none;
}
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}

/* Button used to open the contact form - fixed at the bottom of the page */
.open-button {
background-color: #555;
color: white;
padding: 16px 20px;
border: none;
cursor: pointer;
opacity: 0.8;
position: fixed;
bottom: 23px;
right: 28px;
width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%,-50%);
width: 450px;
padding: 40px;
background: rgba(0,0,0,.8);
box-sizing: border-box;
box-shadow: 0 15px 25px rgba(0,0,0,.5);
border-radius: 10px;
}

/* Add styles to the form container */
.form-container {
max-width: 500px;
padding: 5px;
}

/* Full-width input fields */
.form-container input[type=text], .form-container input[type=password] {
width: 100%;
padding: 14px;
margin: 3px 0 22px 0;
border: none;
background: #f1f1f1;

```

```
}  
  
/* When the inputs get focus, do something */  
.form-container input[type=text]:focus, .form-container input[type=password]:focus {  
  background-color: #ddd;  
  outline: none;  
}  
  
/* Set a style for the submit/login button */  
.form-container .btn {  
  background-color: #4CAF50;  
  color: white;  
  padding: 10px 20px;  
  border: none;  
  cursor: pointer;  
  width: 100%;  
  margin-bottom: 10px;  
  opacity: 0.8;  
}  
  
/* Add a red background color to the cancel button */  
.form-container .cancel {  
  background-color: red;  
}  
  
/* Add some hover effects to buttons */  
.form-container .btn:hover, .open-button:hover {  
  opacity: 1;  
}
```

THE END