Logbook

A PERSONAL TASK MANAGEMENT WEB APPLICATION

PARADOXICAL GALEEB

Submitted to

Emon Kumar Dey
Assistant Professor

Saeed Siddik
Lecturer

Institute of Information Technology
University of Dhaka

Submitted by

Team Pirates

Jahidul Islam (BSSE 0702)

Asadullah Hill Galib (BSSE 0712)

Supervised by

Emon Kumar Dey

Assistant Professor

Institute of Information Technology
University of Dhaka

Session: 2014-2015

Institute of Information Technology

University of Dhaka

Institute of Information Technology

University of Dhaka

30-04-2017

Table of Contents

List of Figures:

Chapter 1: Project Definition

In Software Project Lab –II we are going present a worthwhile and easy-to-use web application – “Logbook”. This project will aim to build a personal task management tool, which can ease our life.

1.1 About the Project

Basically, Project managers are aware of the difficulty in keeping track of various tasks, resources, timeline in maintaining daily life easily. Our proposed personal task manager is such kind of tool which will be serviceable and easy-to-use. It will have some light features to keep track of daily activities:

- Authentication the access
- Add note
- Add attachments (files or images)
- Archive note
- Add Tags
- Search using tags
- Reminder
- Synchronize with Calendar and clock
- Synchronize with mail

1.2 Scope of Our Project

Scope of our project is-

- Mobility of the web application can’t be properly handled
- There will be no retrieval of data if database gets lost or deleted or damaged
- Email Verification is out of scope of this project

Chapter 2: Implementation Overview

This chapter aims to describe the implementation process of “Logbook”. Here the technologies that have been used to develop this system and the testing that have been done during this system development will be described in brief.

2.1 Problem Description

There is are two interfaces- admin interface and user interface.

Users have to authenticate to the system to perform tasks. They can add, remove, update notes, lists, tasks, to-do-list, reminder. They can update their information in the account. There is a common introduction page for all the users. From there they can sign up and log in. A non-user can register to be a user. There is a search-bar to search specific note by using heading or tag words.

2.2 Technology Used in implementation

Development technologies are growing very rapidly with the increase of requirements. The technologies that have been used to develop this system is the most recent technologies and also very much appropriate to it.

2.2.1 Client-side Technology

- **Hyper Text Markup Language (HTML)**

Hyper Text Markup Language (HTML) is the main markup language for web pages. HTML elements are the basic building-blocks of a webpage. We have used the latest HTML5 for developing this system.

- **Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML. We have used css3 for developing this system. CSS3 is the latest standard for CSS. It is completely backwards-compatible with earlier versions of CSS.

- **JavaScript(JS)**

JavaScript is a high-level, dynamic and interpreted run-time language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. The majority of websites employ it and all modern Web browsers support it without the need for plug-ins. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

- **W3-CSS**

W3.CSS is a modern CSS framework with built-in responsiveness.

- Smaller and faster than other CSS frameworks.
- Easier to learn, and easier to use than other CSS frameworks.
- Uses standard CSS only (No jQuery or JavaScript library).
- Speeds up and simplifies web development.
- Supports modern responsive design (mobile first) by default.
- Provides CSS equality for all browsers. Chrome, Firefox, IE, Safari, and more.

- Provides CSS equality for all devices. PC, laptop, tablet, and mobile

- **Bootstrap (front-end framework)**

Bootstrap is a free and open-source collection of tools for creating websites and web applications.

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

2.2.2 Server-side Technology

For back-end coding, we used various languages and libraries-

- **PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

• MYSQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase

this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

2.2.3 Implementation Tools

As types of software are increasing day by day, new implementation tools are also needed for their implementation. Nowadays, there are many implementation tools. Developers have to choose right tools for each part of their application. If they can utilize tools perfectly, their labor can be reduced.

- **Apache HTTP Server**

The Apache HTTP Server, colloquially called Apache is the world's most used web server software. Released under the Apache License, Apache is free and open-source software. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes.

- **XAMPP Control Panel v3.2.2**

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). 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. Major features as available in XAMPP:

- It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.
- Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.
- XAMPP is cross-platform, it works equally well on Linux, Mac and Windows.
- Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Chapter 3: Source Code Description

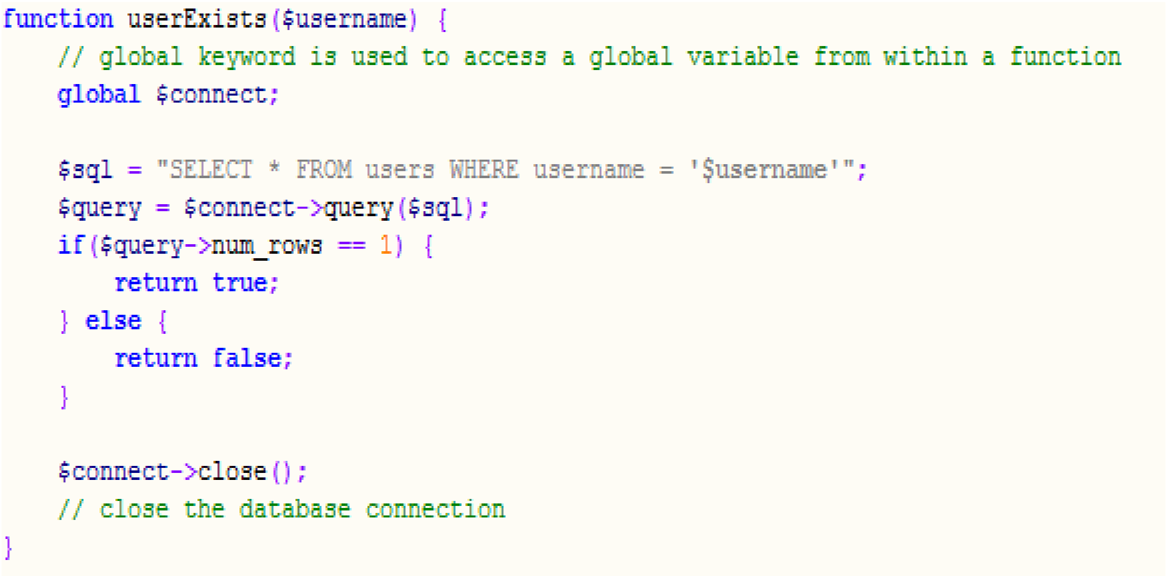
We can divide user manual into two section, Non-User and User

3.1 Non-User

There are two major part of this side. They are- sign up, log in.

3.1.1 Sign Up

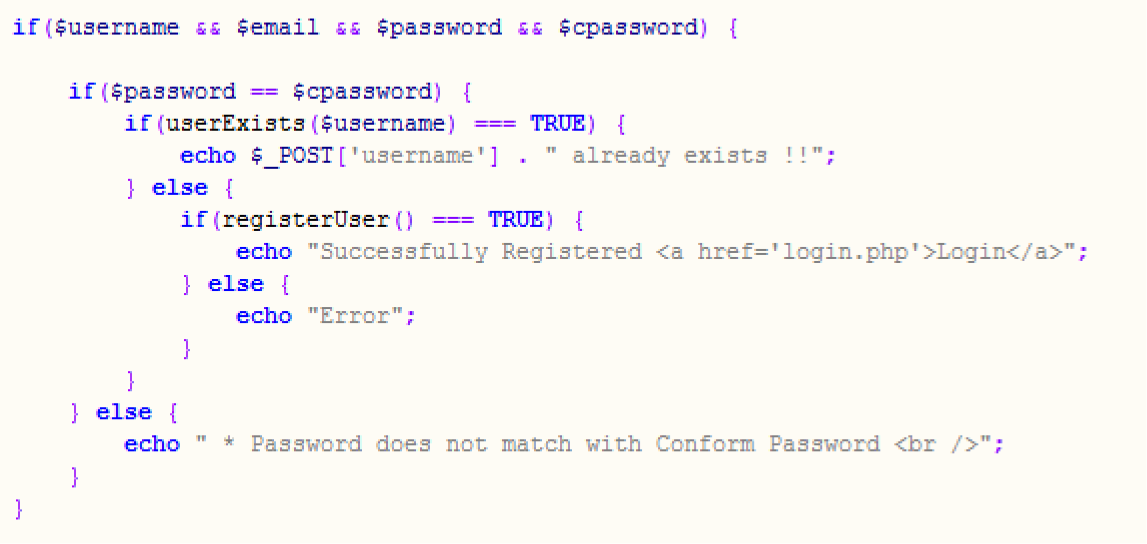
- Checking if username already exists



```
function userExists($username) {
    // global keyword is used to access a global variable from within a function
    global $connect;

    $sql = "SELECT * FROM users WHERE username = '$username'";
    $query = $connect->query($sql);
    if($query->num_rows == 1) {
        return true;
    } else {
        return false;
    }

    $connect->close();
    // close the database connection
}
```

Figure 1: Code: SignUp(1)

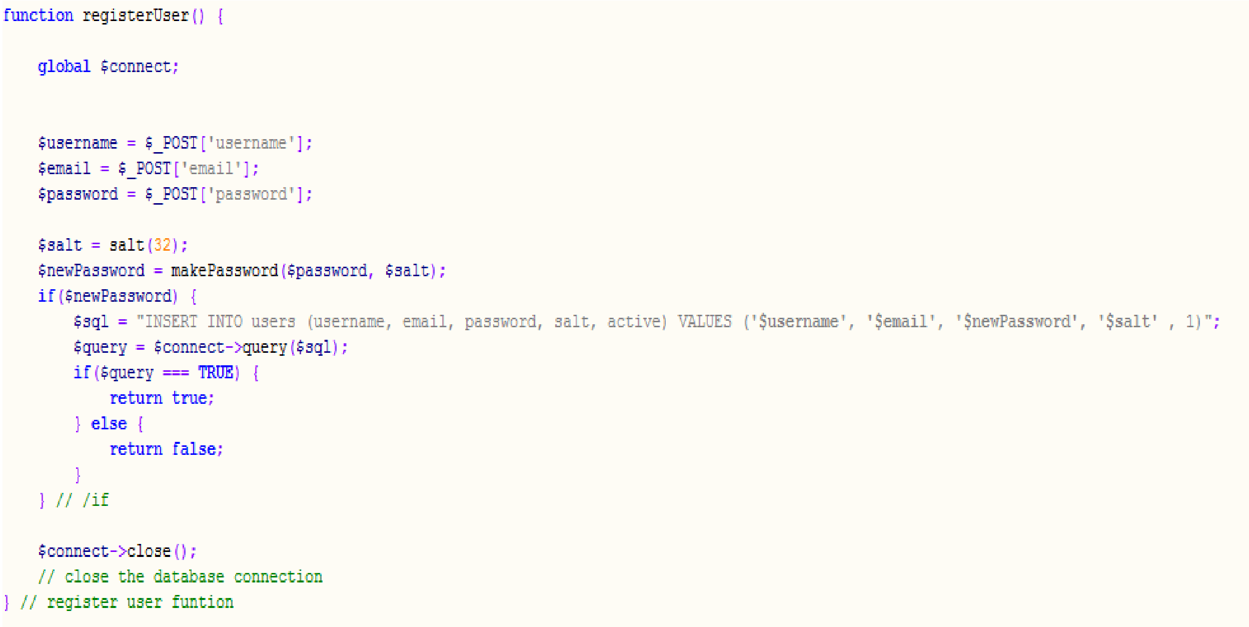
- Matching two password



```
if($username && $email && $password && $cpassword) {

    if($password == $cpassword) {
        if(userExists($username) === TRUE) {
            echo $_POST['username'] . " already exists !!";
        } else {
            if(registerUser() === TRUE) {
                echo "Successfully Registered <a href='login.php'>Login</a>";
            } else {
                echo "Error";
            }
        }
    } else {
        echo " * Password does not match with Conform Password <br />";
    }
}
```

Figure 2: Code: SignUp(2)

- Register user and insert info. into database.

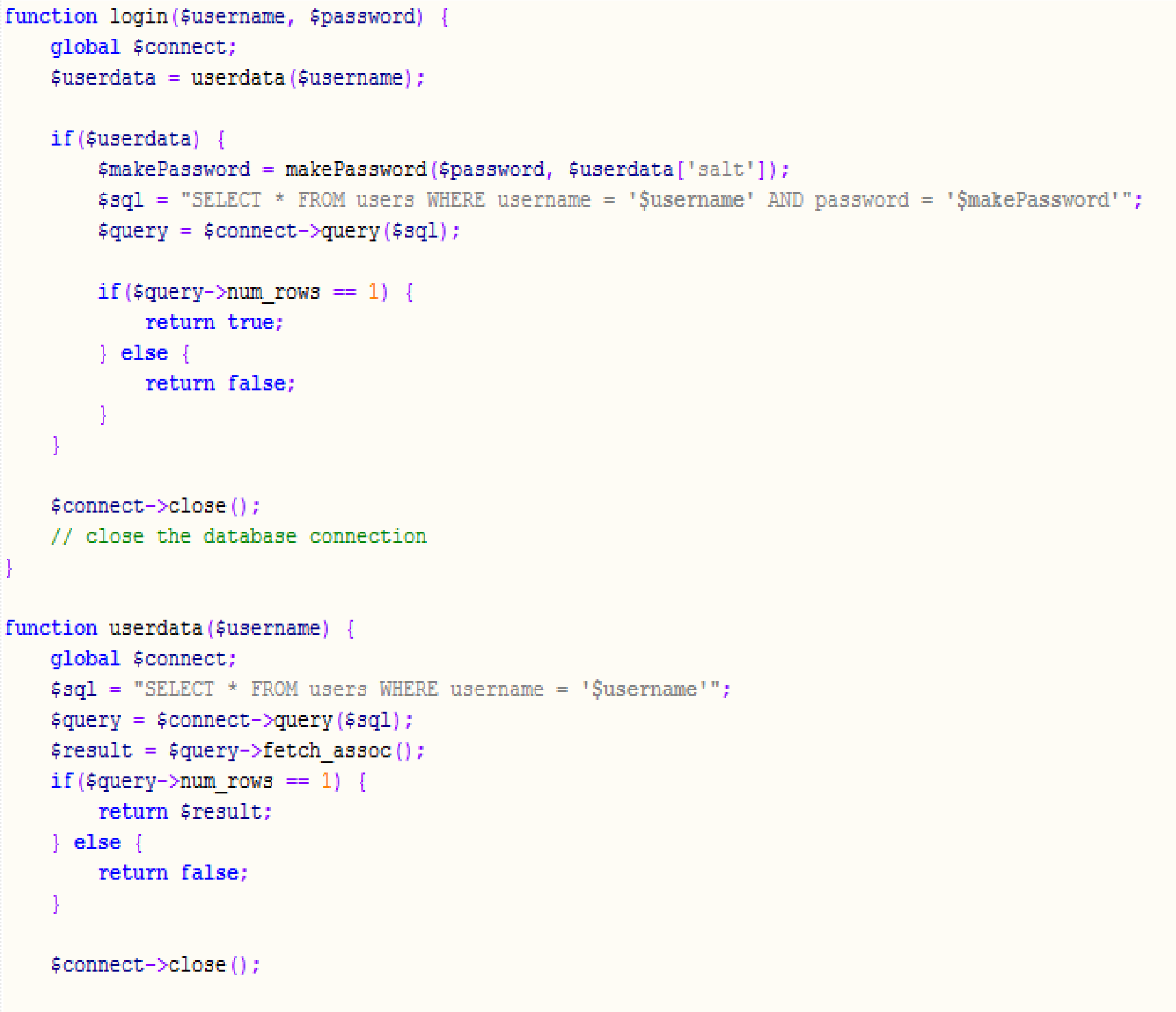


```
function registerUser() {

    global $connect;


    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    $salt = salt(32);
    $newPassword = makePassword($password, $salt);
    if($newPassword) {
        $sql = "INSERT INTO users (username, email, password, salt, active) VALUES ('$username', '$email', '$newPassword', '$salt' , 1)";
        $query = $connect->query($sql);
        if($query === TRUE) {
            return true;
        } else {
            return false;
        }
    } // /if

    $connect->close();
    // close the database connection
} // register user funtion
```

Figure 3: Code: SignUp(3)

3.1.2 Log In

- First checking whether the username exists or not? Code snippet for this section is same as sign up section's username exists part.
- Then fetching the row according to username from database and match the password. Of course the encrypted password is used here, for security issue.



```
function login($username, $password) {
    global $connect;
    $userdata = userdata($username);

    if($userdata) {
        $makePassword = makePassword($password, $userdata['salt']);
        $sql = "SELECT * FROM users WHERE username = '$username' AND password = '$makePassword'";
        $query = $connect->query($sql);

        if($query->num_rows == 1) {
            return true;
        } else {
            return false;
        }
    }

    $connect->close();
    // close the database connection
}

function userdata($username) {
    global $connect;
    $sql = "SELECT * FROM users WHERE username = '$username'";
    $query = $connect->query($sql);
    $result = $query->fetch_assoc();
    if($query->num_rows == 1) {
        return $result;
    } else {
        return false;
    }

    $connect->close();
```

Figure 4: Code: LogIn(1)

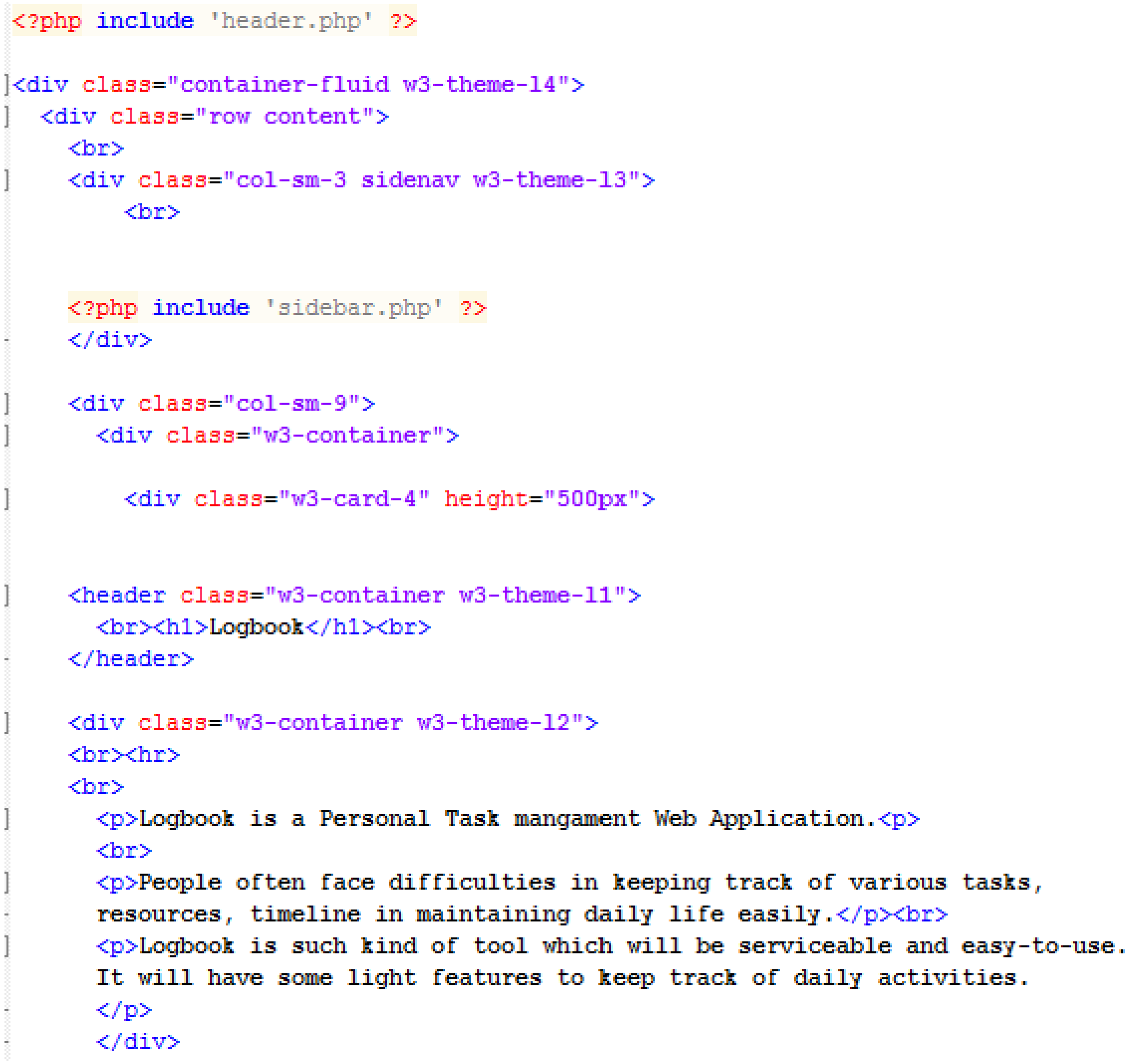
- Then proceed to home page or show password mismatch massage.

3.2 User

User can perform multiple task after get logged in.

3.2.1 Home

- This page contains an introductory w3-card.

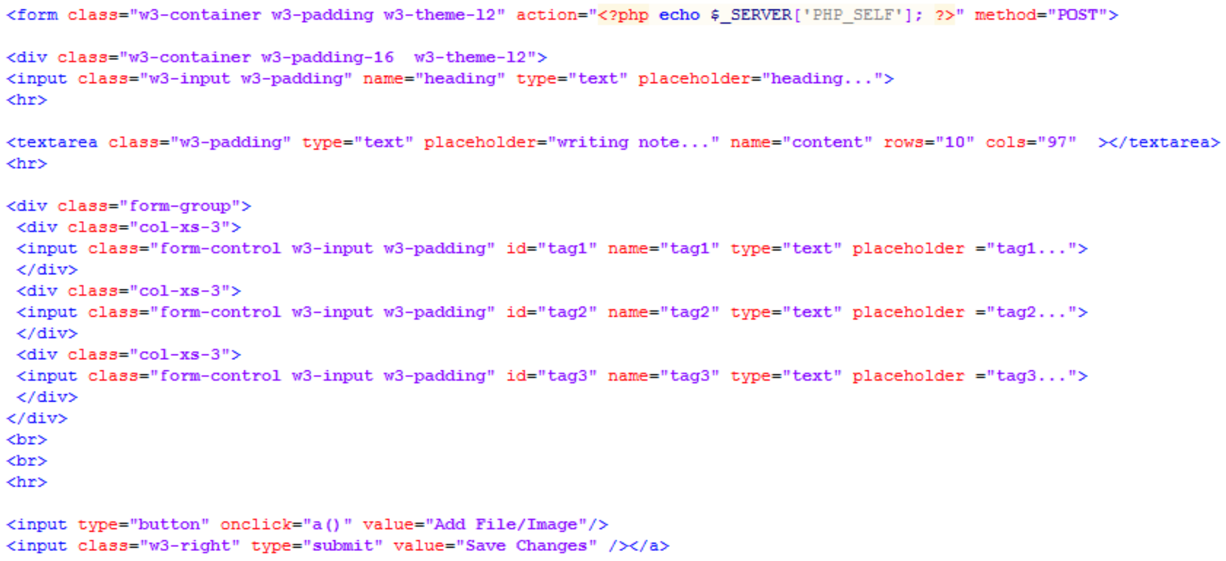


```
<?php include 'header.php' ?>

<div class="container-fluid w3-theme-14">
  <div class="row content">
    <br>
    <div class="col-sm-3 sidenav w3-theme-13">
        <br>


    <?php include 'sidebar.php' ?>
    </div>

    <div class="col-sm-9">
      <div class="w3-container">

        <div class="w3-card-4" height="500px">


    <header class="w3-container w3-theme-11">
      <br><h1>Logbook</h1><br>
    </header>

    <div class="w3-container w3-theme-12">
    <br><hr>
    <br>
      <p>Logbook is a Personal Task mangament Web Application.<p>
      <br>
      <p>People often face difficulties in keeping track of various tasks,
      resources, timeline in maintaining daily life easily.</p><br>
      <p>Logbook is such kind of tool which will be serviceable and easy-to-use.
      It will have some light features to keep track of daily activities.
      </p>
      </div>
```

Figure 5: Code: Home

3.2.2 Adding Note Page

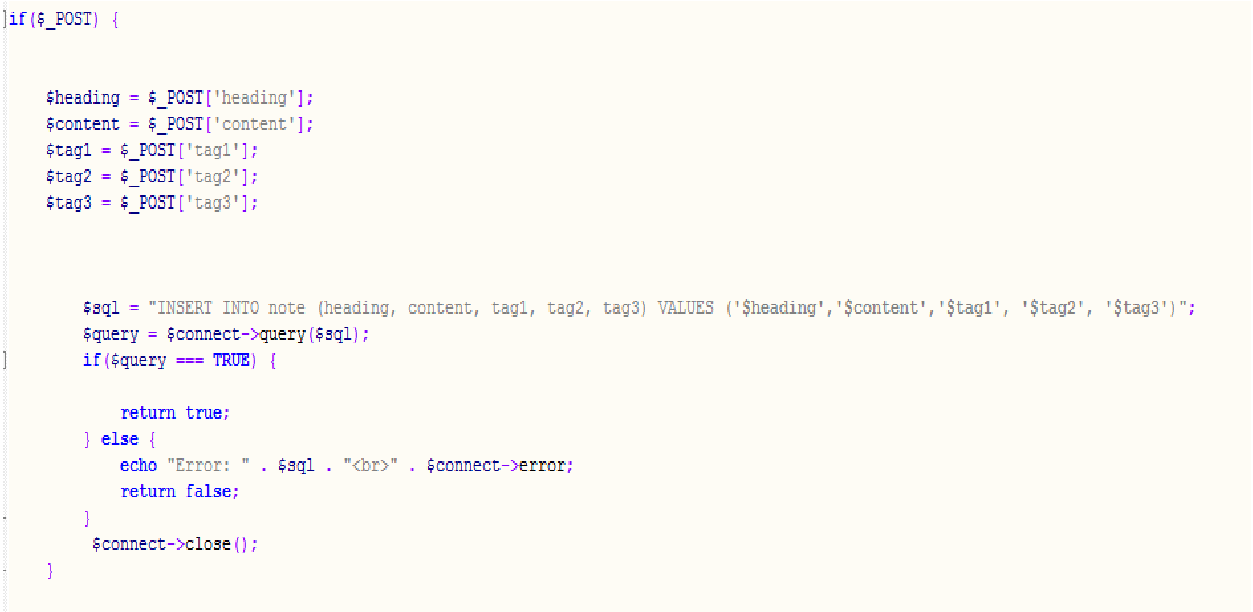
- User fill up the content of the note form like heading, content, tag1, tag2, tag3, adding attachment



```
<form class="w3-container w3-padding w3-theme-l2" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">

<div class="w3-container w3-padding-16  w3-theme-l2">
<input class="w3-input w3-padding" name="heading" type="text" placeholder="heading...">
<hr>

<textarea class="w3-padding" type="text" placeholder="writing note..." name="content" rows="10" cols="97"  ></textarea>
<hr>

<div class="form-group">
 <div class="col-xs-3">
 <input class="form-control w3-input w3-padding" id="tag1" name="tag1" type="text" placeholder ="tag1...">
 </div>
 <div class="col-xs-3">
 <input class="form-control w3-input w3-padding" id="tag2" name="tag2" type="text" placeholder ="tag2...">
 </div>
 <div class="col-xs-3">
 <input class="form-control w3-input w3-padding" id="tag3" name="tag3" type="text" placeholder ="tag3...">
 </div>
</div>
<br>
<br>
<hr>

<input type="button" onclick="a()" value="Add File/Image"/>
<input class="w3-right" type="submit" value="Save Changes" /></a>
```

Figure 6: Code: Adding Note(1)

- After submitting the form by clicking 'save changes' the input will be stored into database.

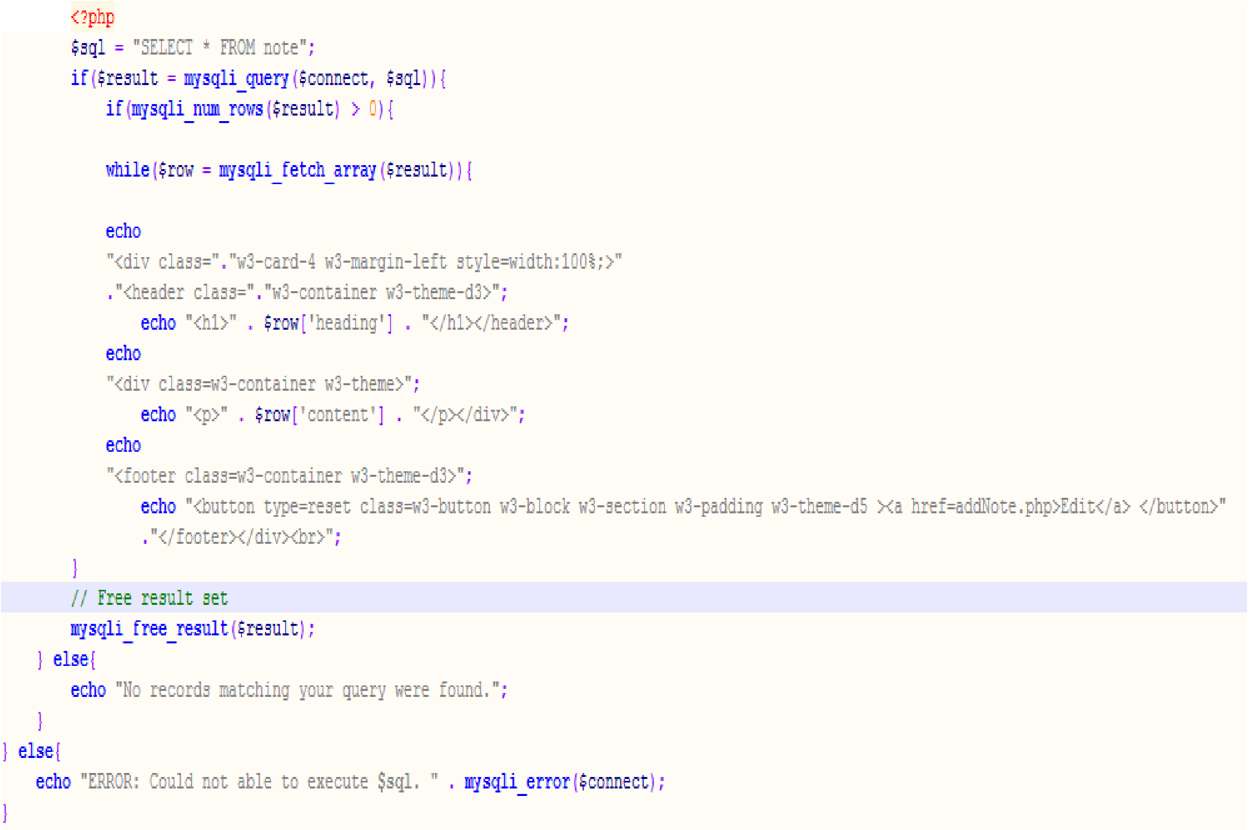


```
if($_POST) {


    $heading = $_POST['heading'];
    $content = $_POST['content'];
    $tag1 = $_POST['tag1'];
    $tag2 = $_POST['tag2'];
    $tag3 = $_POST['tag3'];



        $sql = "INSERT INTO note (heading, content, tag1, tag2, tag3) VALUES ('$heading','$content','$tag1', '$tag2', '$tag3')";
        $query = $connect->query($sql);
        if($query === TRUE) {

            return true;
        } else {
            echo "Error: " . $sql . "<br>" . $connect->error;
            return false;
        }
         $connect->close();
    }
```

Figure 7: Code: Adding Note(2)

3.2.3 View Note Page

- Retrieving ‵note‵ table data from database and showing them simultaneously to the user.
- Showing individual note into w3-card
- Showing note’s heading and content only
- And adding a reference to edit the particular note

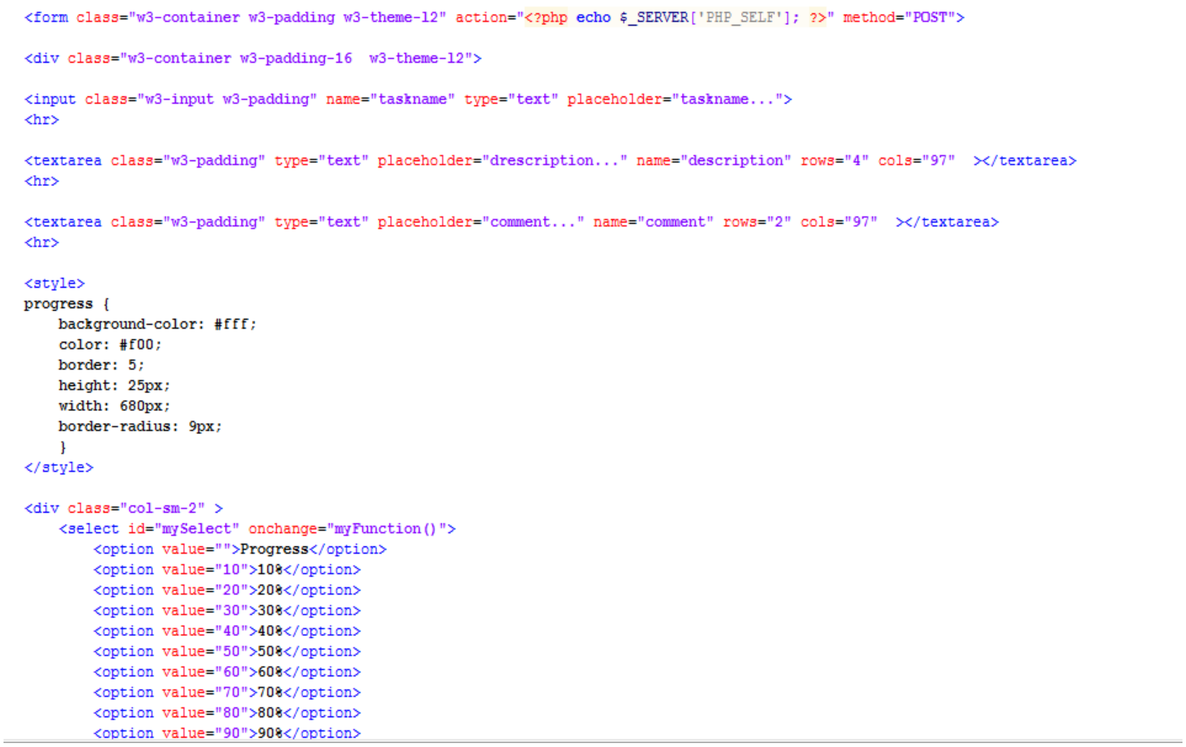


```
        <?php
        $sql = "SELECT * FROM note";
        if($result = mysqli_query($connect, $sql)){
            if(mysqli_num_rows($result) > 0){

            while($row = mysqli_fetch_array($result)){

            echo
            "<div class="."w3-card-4 w3-margin-left style=width:100%;>"
            ."<header class="."w3-container w3-theme-d3>";
                echo "<h1>" . $row['heading'] . "</h1></header>";
            echo
            "<div class=w3-container w3-theme>";
                echo "<p>" . $row['content'] . "</p></div>";
            echo
            "<footer class=w3-container w3-theme-d3>";
                echo "<button type=reset class=w3-button w3-block w3-section w3-padding w3-theme-d5 ><a href=addNote.php>Edit</a> </button>"
                ."</footer></div><br>";
        }
        // Free result set
        mysqli_free_result($result);
    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($connect);
}
```

Figure 8: Code: View Note

3.2.4 Adding Task Page

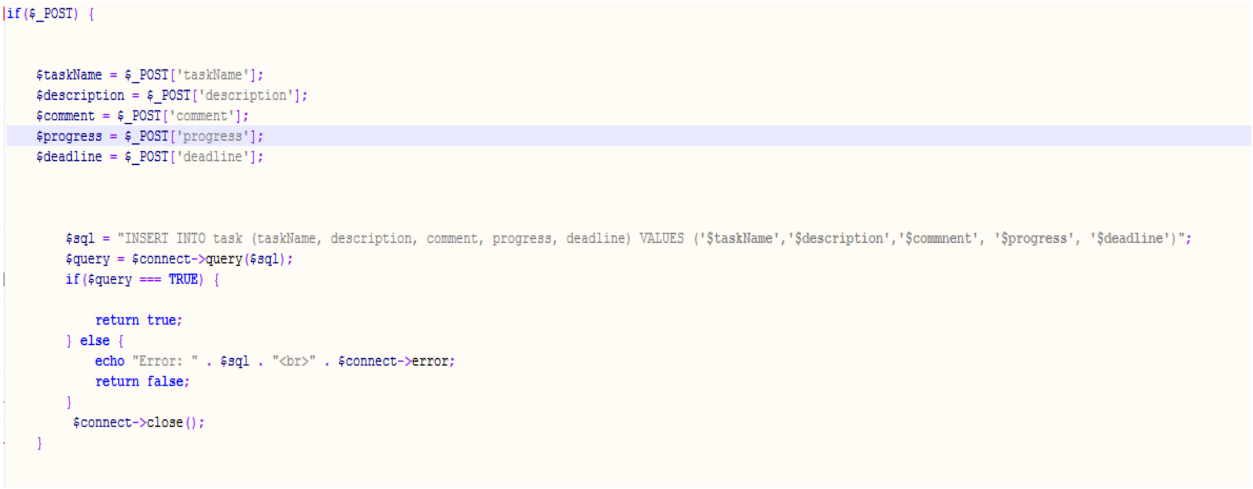
- User fill up the content of the task form like as taskName, description, comment, progress, deadline



```
<form class="w3-container w3-padding w3-theme-12" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">

<div class="w3-container w3-padding-16  w3-theme-12">

<input class="w3-input w3-padding" name="taskname" type="text" placeholder="taskname...">
<hr>

<textarea class="w3-padding" type="text" placeholder="drescription..." name="description" rows="4" cols="97"  ></textarea>
<hr>

<textarea class="w3-padding" type="text" placeholder="comment..." name="comment" rows="2" cols="97"  ></textarea>
<hr>

<style>
progress {
    background-color: #fff;
    color: #f00;
    border: 5;
    height: 25px;
    width: 680px;
    border-radius: 9px;
    }
</style>

<div class="col-sm-2" >
    <select id="mySelect" onchange="myFunction()">
        <option value="">Progress</option>
        <option value="10">10%</option>
        <option value="20">20%</option>
        <option value="30">30%</option>
        <option value="40">40%</option>
        <option value="50">50%</option>
        <option value="60">60%</option>
        <option value="70">70%</option>
        <option value="80">80%</option>
        <option value="90">90%</option>
```

Figure 9: Code: Adding Task(1)

- After submitting the form by clicking 'save changes' the input will be stored into database.

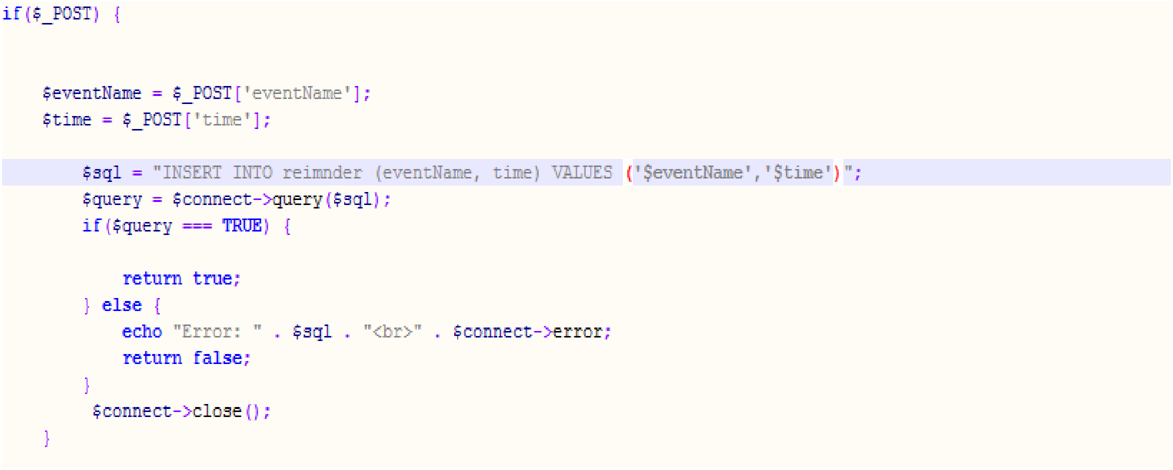


```
if($_POST) {


    $taskName = $_POST['taskName'];
    $description = $_POST['description'];
    $comment = $_POST['comment'];
    $progress = $_POST['progress'];
    $deadline = $_POST['deadline'];



        $sql = "INSERT INTO task (taskName, description, comment, progress, deadline) VALUES ('$taskName','$description','$commnent', '$progress', '$deadline')";
        $query = $connect->query($sql);
        if($query === TRUE) {

            return true;
        } else {
            echo "Error: " . $sql . "<br>" . $connect->error;
            return false;
        }
         $connect->close();
    }
```

Figure 10: Code: Adding Task(2)

3.2.5 Adding Reminder Page

- Setting Reminder’s event name and time and updating database

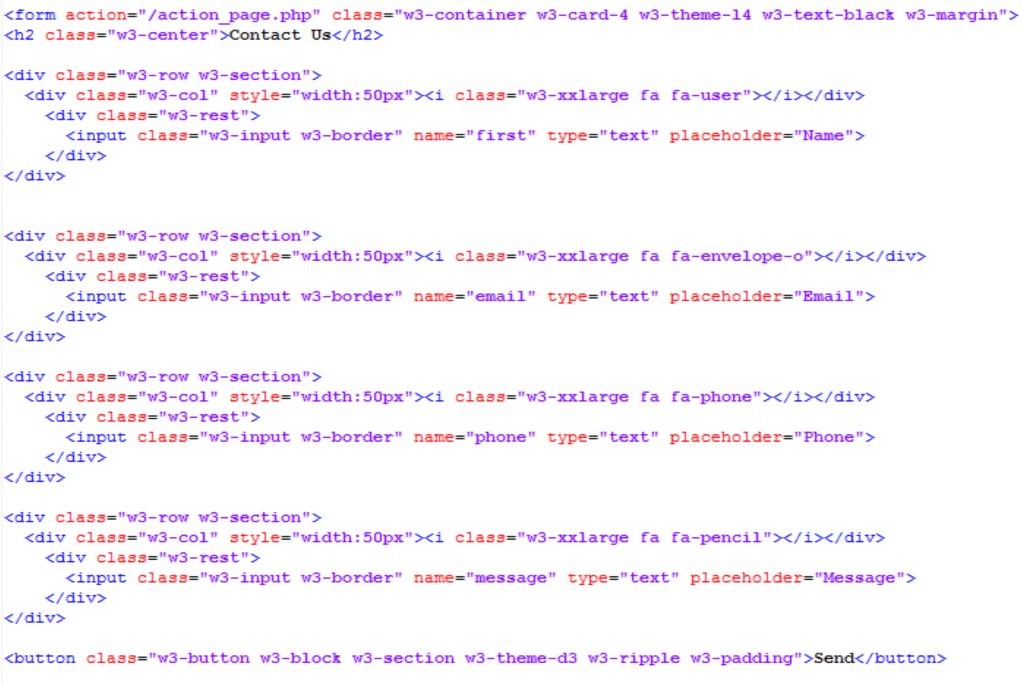


```
if($_POST) {


    $eventName = $_POST['eventName'];
    $time = $_POST['time'];

        $sql = "INSERT INTO reimnder (eventName, time) VALUES ('$eventName','$time')";
        $query = $connect->query($sql);
        if($query === TRUE) {

            return true;
        } else {
            echo "Error: " . $sql . "<br>" . $connect->error;
            return false;
        }
         $connect->close();
    }
```

Figure 11: Code: Adding Reminder

3.2.6 Contact Page

- User can directly send any complain or advise to the server side controller by this part.

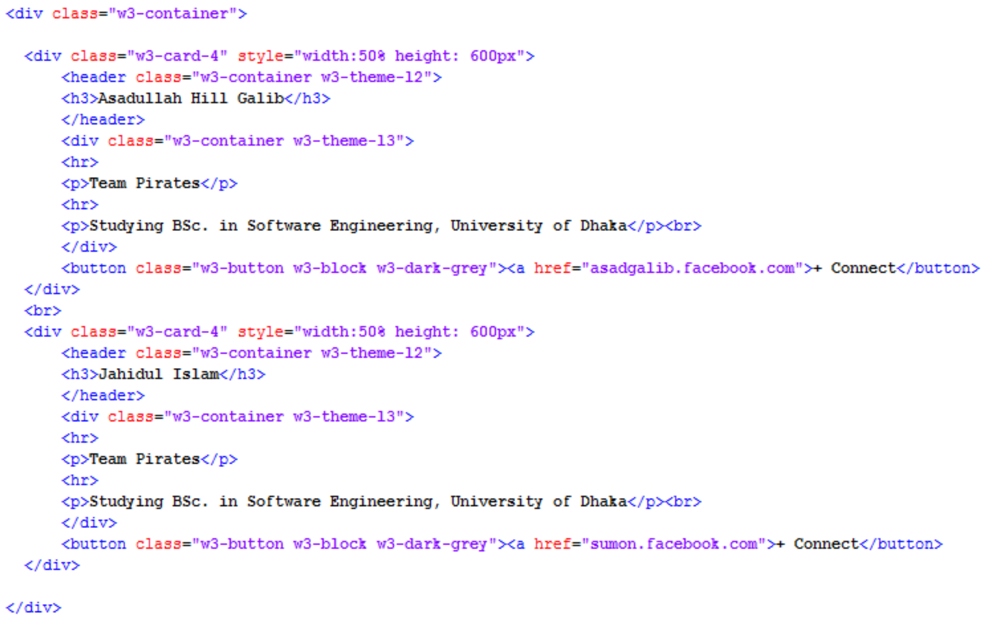


```
<form action="/action_page.php" class="w3-container w3-card-4 w3-theme-l4 w3-text-black w3-margin">
<h2 class="w3-center">Contact Us</h2>

<div class="w3-row w3-section">
  <div class="w3-col" style="width:50px"><i class="w3-xxlarge fa fa-user"></i></div>
    <div class="w3-rest">
      <input class="w3-input w3-border" name="first" type="text" placeholder="Name">
    </div>
</div>


<div class="w3-row w3-section">
  <div class="w3-col" style="width:50px"><i class="w3-xxlarge fa fa-envelope-o"></i></div>
    <div class="w3-rest">
      <input class="w3-input w3-border" name="email" type="text" placeholder="Email">
    </div>
</div>

<div class="w3-row w3-section">
  <div class="w3-col" style="width:50px"><i class="w3-xxlarge fa fa-phone"></i></div>
    <div class="w3-rest">
      <input class="w3-input w3-border" name="phone" type="text" placeholder="Phone">
    </div>
</div>

<div class="w3-row w3-section">
  <div class="w3-col" style="width:50px"><i class="w3-xxlarge fa fa-pencil"></i></div>
    <div class="w3-rest">
      <input class="w3-input w3-border" name="message" type="text" placeholder="Message">
    </div>
</div>

<button class="w3-button w3-block w3-section w3-theme-d3 w3-ripple w3-padding">Send</button>
```

Figure 12: Code: Contact

3.2.7 About Us Page

- This page conatains a short description of the developers.

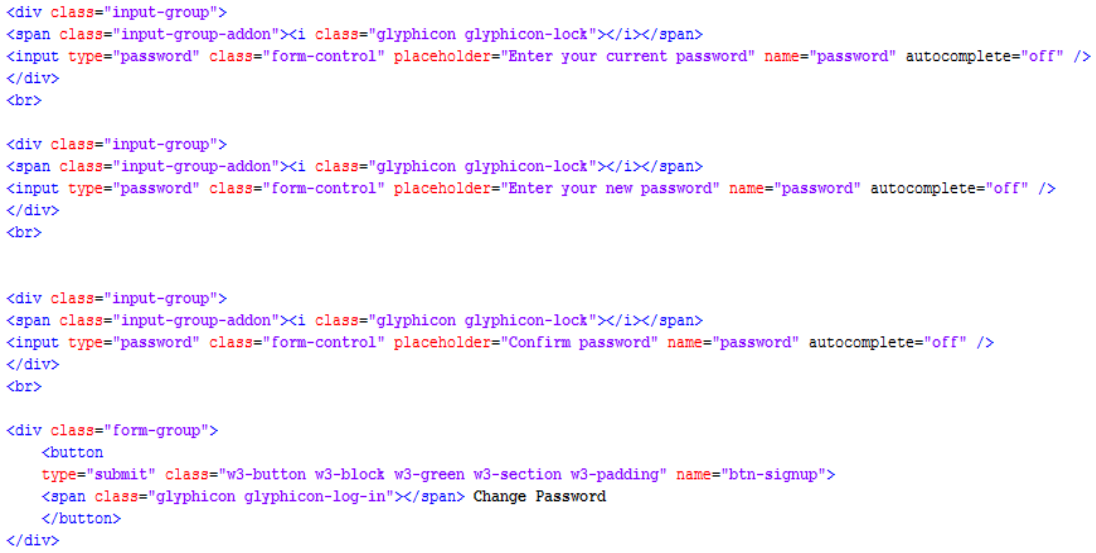


```
<div class="w3-container">

  <div class="w3-card-4" style="width:50% height: 600px">
      <header class="w3-container w3-theme-l2">
      <h3>Asadullah Hill Galib</h3>
      </header>
      <div class="w3-container w3-theme-l3">
      <hr>
      <p>Team Pirates</p>
      <hr>
      <p>Studying BSc. in Software Engineering, University of Dhaka</p><br>
      </div>
      <button class="w3-button w3-block w3-dark-grey"><a href="asadgalib.facebook.com">+ Connect</button>
  </div>
  <br>
  <div class="w3-card-4" style="width:50% height: 600px">
      <header class="w3-container w3-theme-l2">
      <h3>Jahidul Islam</h3>
      </header>
      <div class="w3-container w3-theme-l3">
      <hr>
      <p>Team Pirates</p>
      <hr>
      <p>Studying BSc. in Software Engineering, University of Dhaka</p><br>
      </div>
      <button class="w3-button w3-block w3-dark-grey"><a href="sumon.facebook.com">+ Connect</button>
  </div>

</div>
```

Figure 13: Code: About Us

3.2.8 Change Password Page

- Here user can change her password

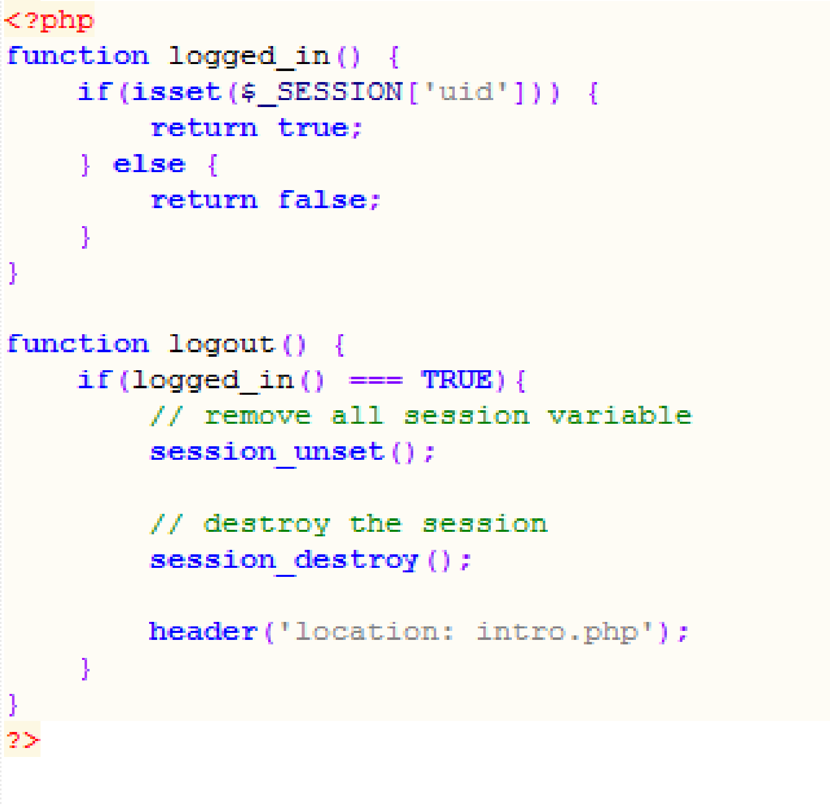


```
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
<input type="password" class="form-control" placeholder="Enter your current password" name="password" autocomplete="off" />
</div>
<br>

<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
<input type="password" class="form-control" placeholder="Enter your new password" name="password" autocomplete="off" />
</div>
<br>


<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
<input type="password" class="form-control" placeholder="Confirm password" name="password" autocomplete="off" />
</div>
<br>

<div class="form-group">
    <button
    type="submit" class="w3-button w3-block w3-green w3-section w3-padding" name="btn-signup">
    <span class="glyphicon glyphicon-log-in"></span> Change Password
    </button>
</div>
```

Figure 14: Code: Changing Password

3.2.9 Log Out

- And Finally User can logged out using log out button.
- Session destroyed and user’s location will be the inro page



```php
<?php
function logged_in() {
    if(isset($_SESSION['uid'])) {
        return true;
    } else {
        return false;
    }
}

function logout() {
    if(logged_in() === TRUE){
        // remove all session variable
        session_unset();

        // destroy the session
        session_destroy();

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

Figure 15: Code: Log Out

Chapter 4: User Manual

We can divide user manual into two section, Non-User and User

4.1 Non-User

There are three major part of this side. They are- intro page, sign up page, log in page.

4.1.1 Intro Page

Both the member and non-member users get this intropage when they first enter to this website.

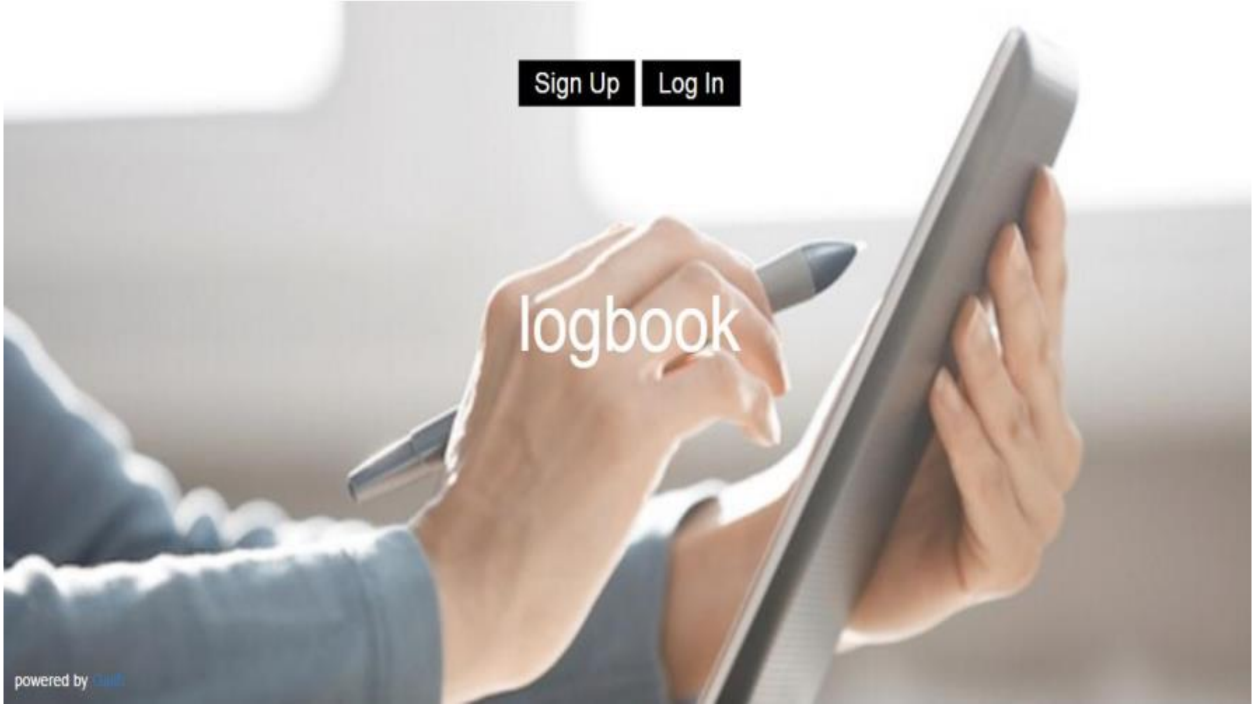
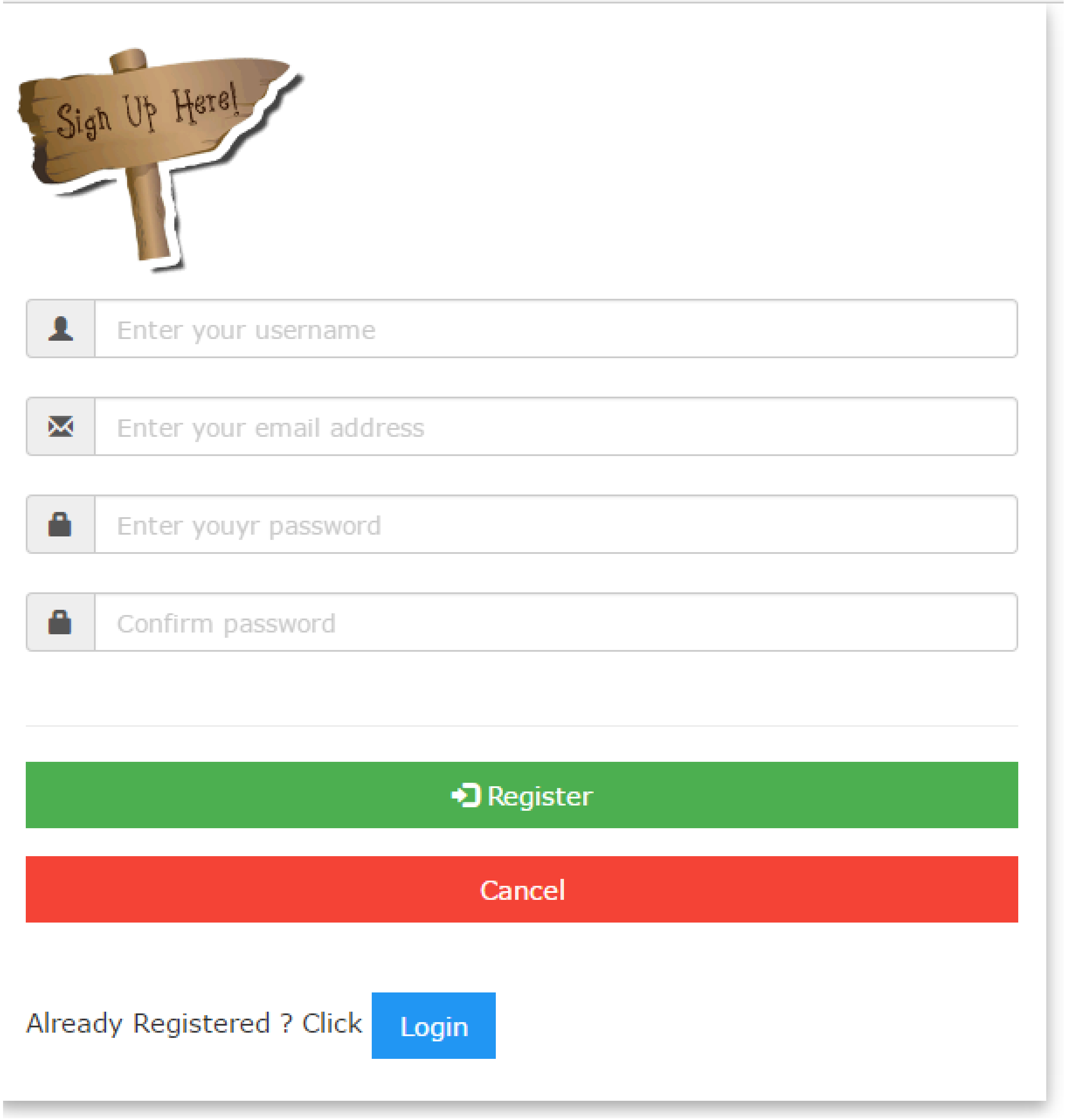


Figure 16: User Manual: Intro

4.1.2 Sign Up Page

By clicking sign up button user can get access to sign up page, where she can provide info and get signed up.



Already Registered ? Click

Figure 17: User Manual: Sign Up

4.1.3 Log In Page

By clicking log in button user can get access to sign up page, where she can provide info and get Logged in.

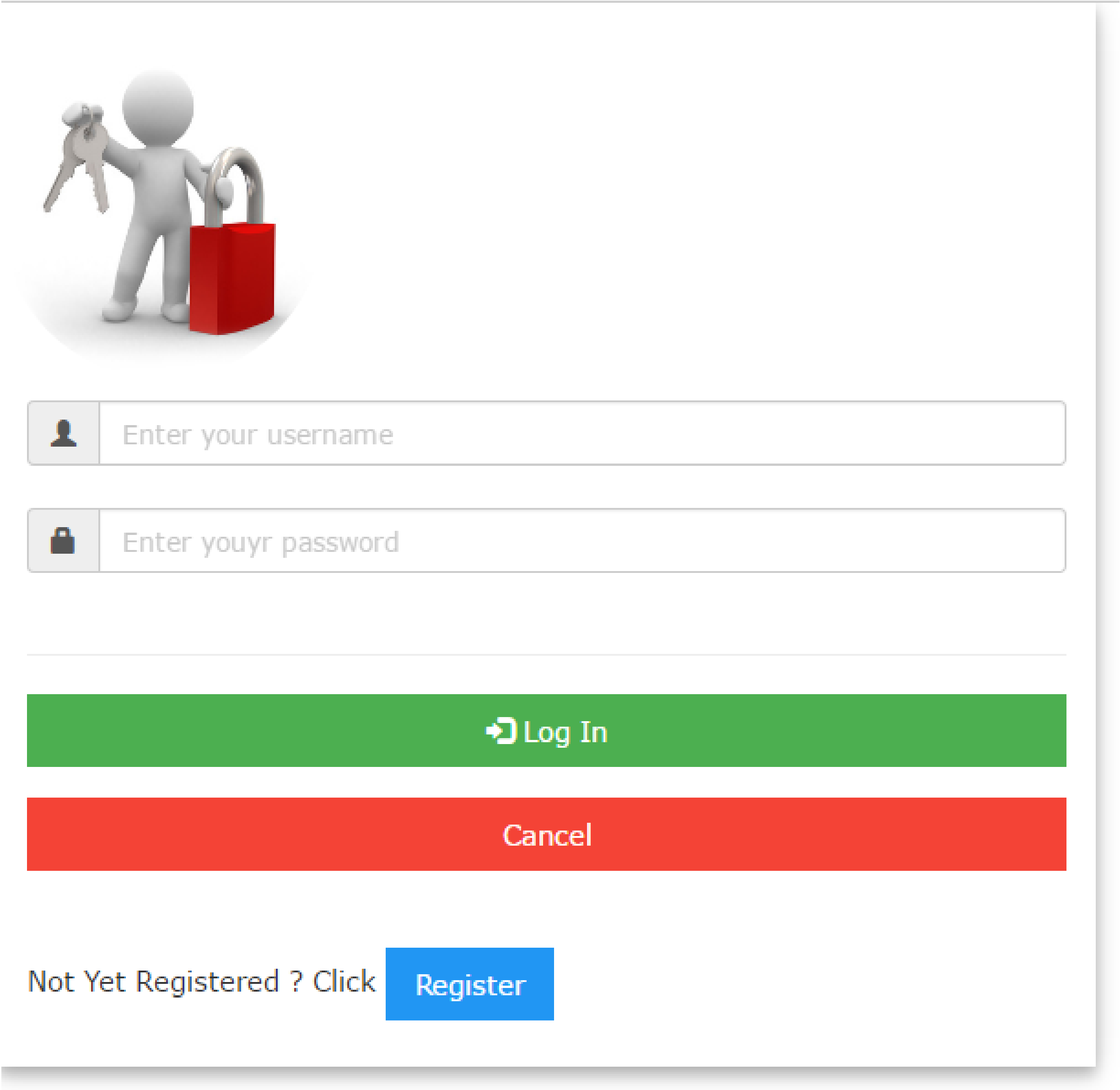


Figure 18: User Manual: Log In

4.2 User

User can perform multiple task after get logged in.

4.2.1 Home Page

After user get authenticated, she will get access to the home page.

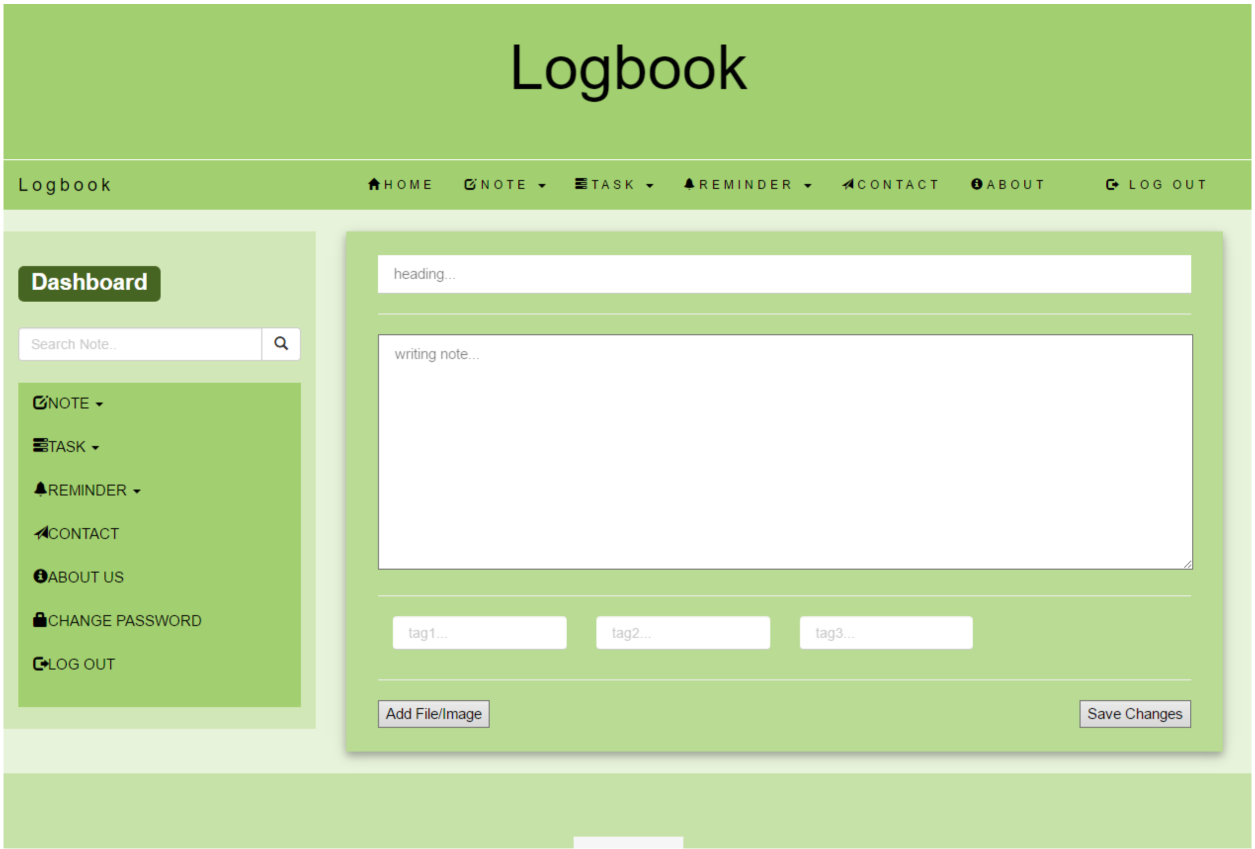


Figure 19: User Manual: Home

4.2.2 Adding Note Page

After selecting add note option, user can write note in text format, add tag, heading and also can attach file/image. And finally she can save changes.

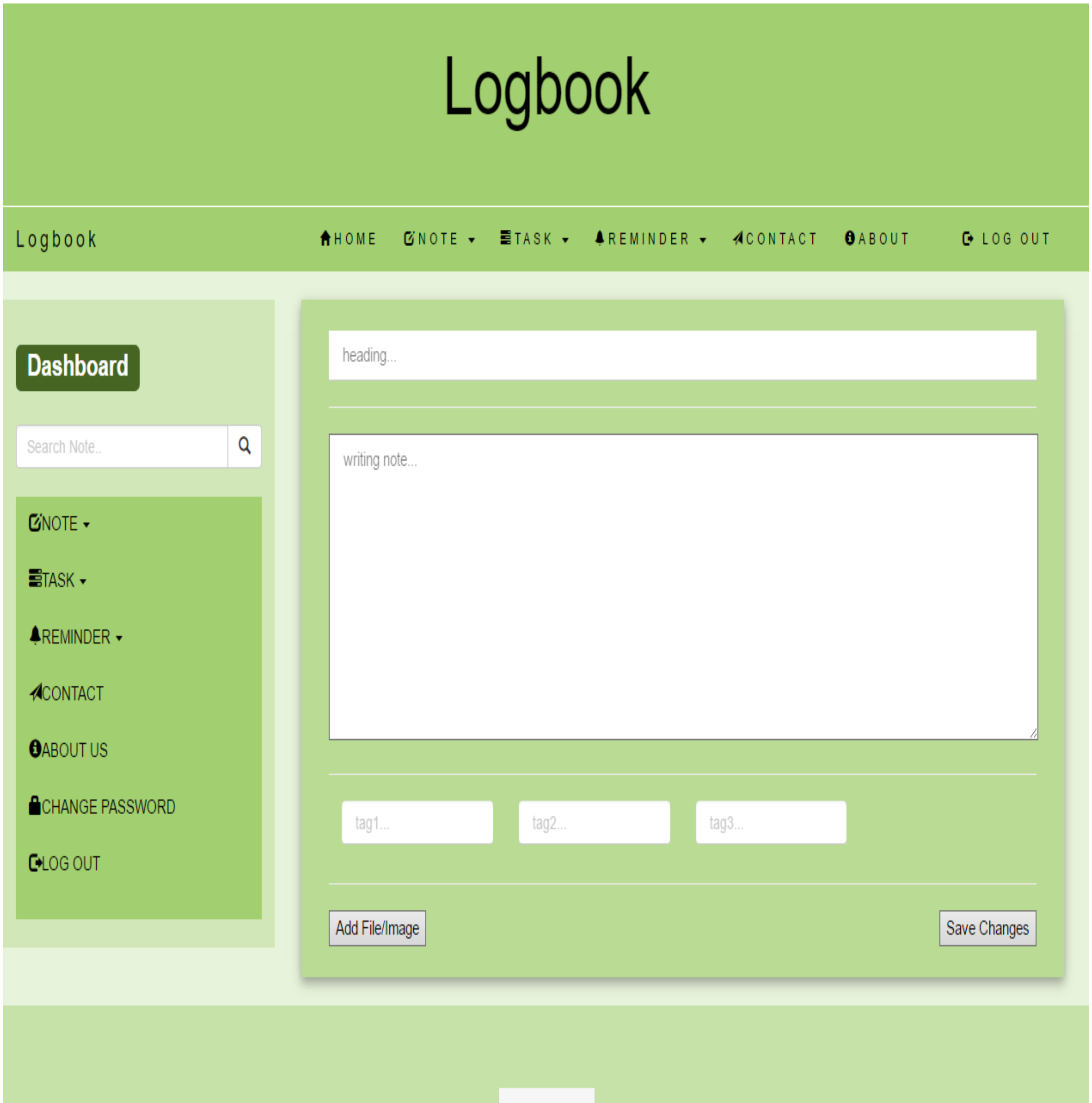
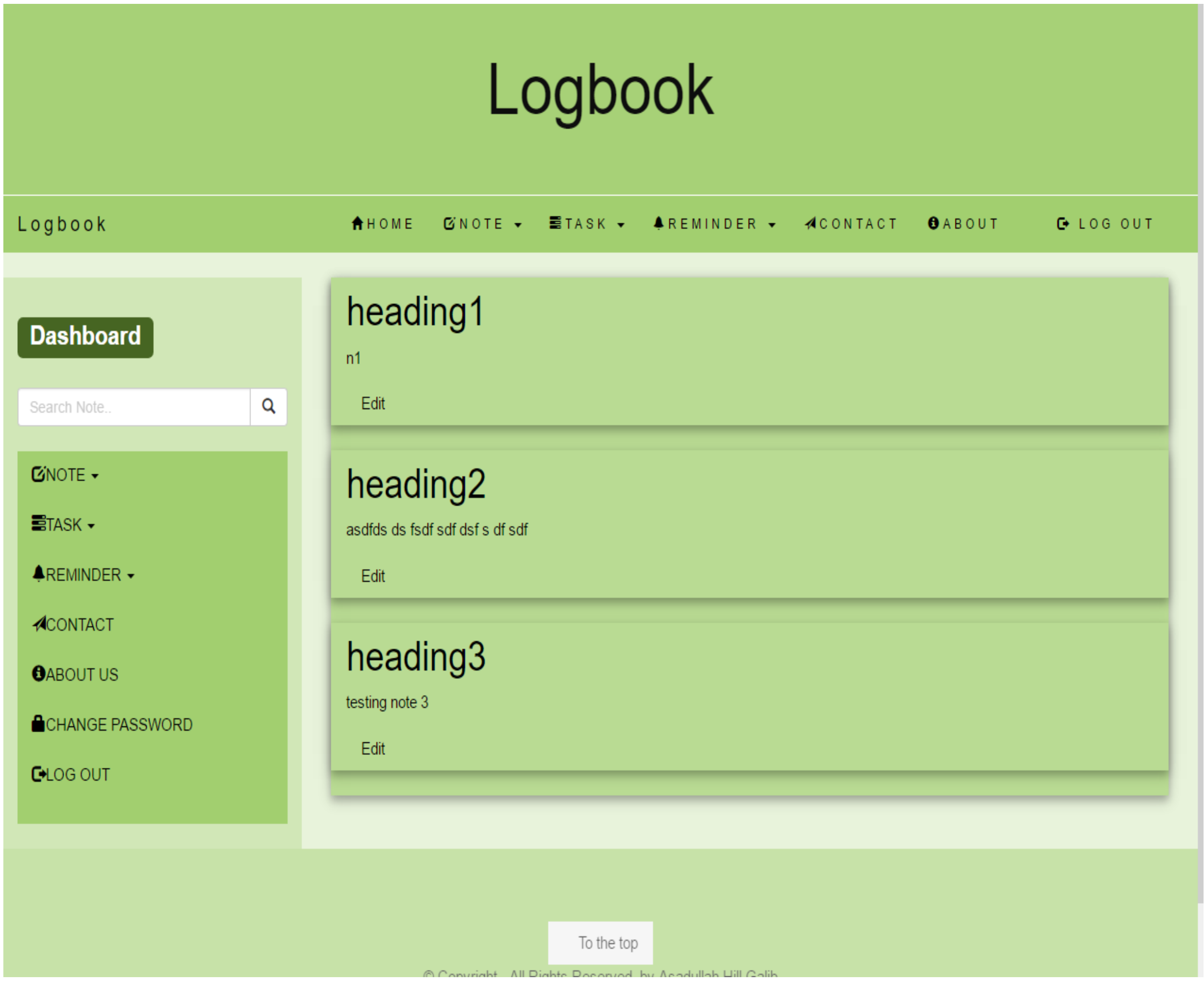


Figure 20: User Manual: Adding Note

4.2.3 View Note Page

After selecting the view note option, user can view previously saved note in text format. She also have a option to edit the note again.



heading1

n1

Edit

heading2

asdfds ds fsdf sdf dsf s df sdf

Edit

heading3

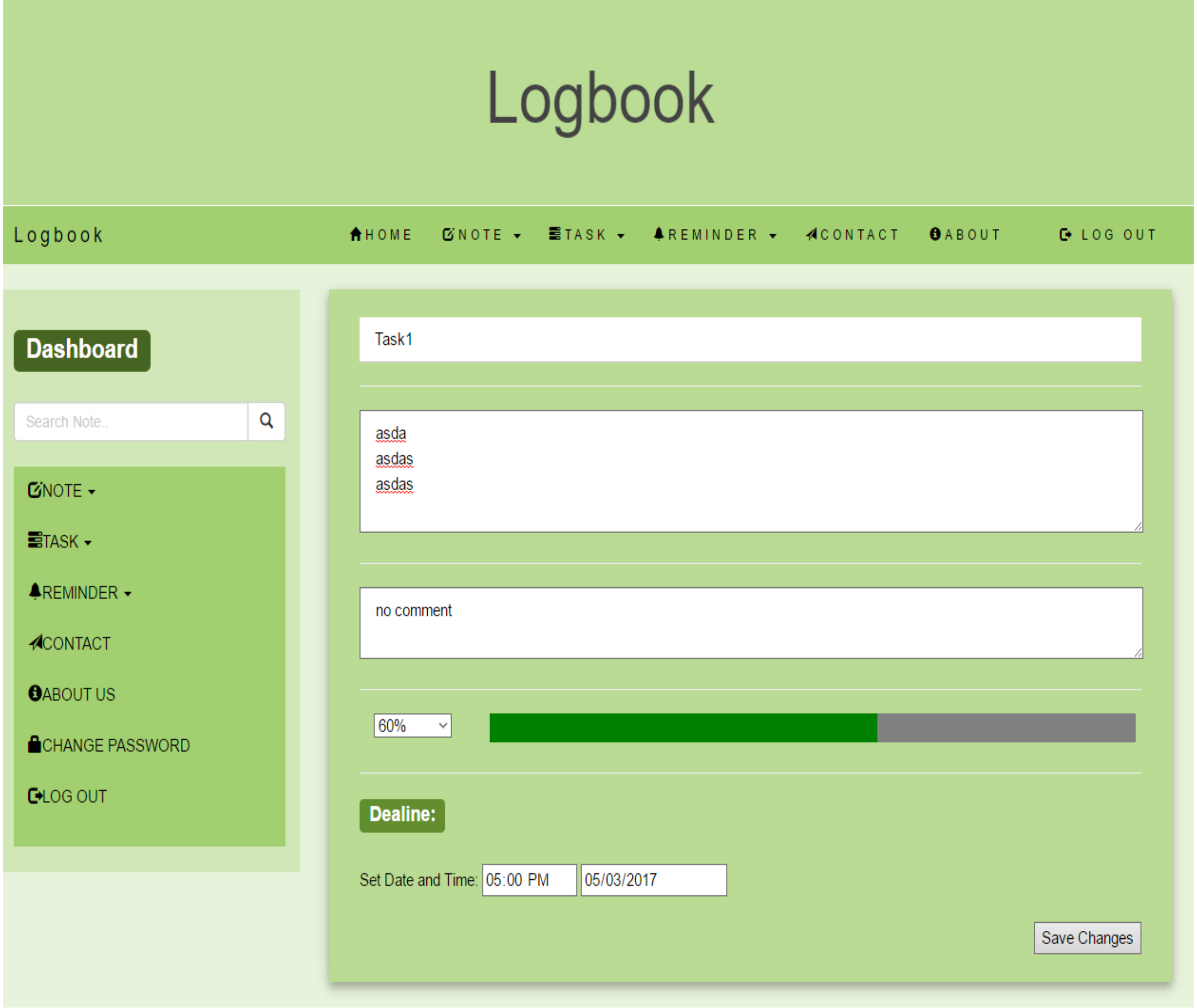
testing note 3

Edit

Figure 21: User Manual: View Note

4.2.4 Adding Task Page

After selecting add task option, user can set a task and also set its description, progress status, deadline. And finally she can save changes.



Task1

asda
asdas
asdas

no comment

60%

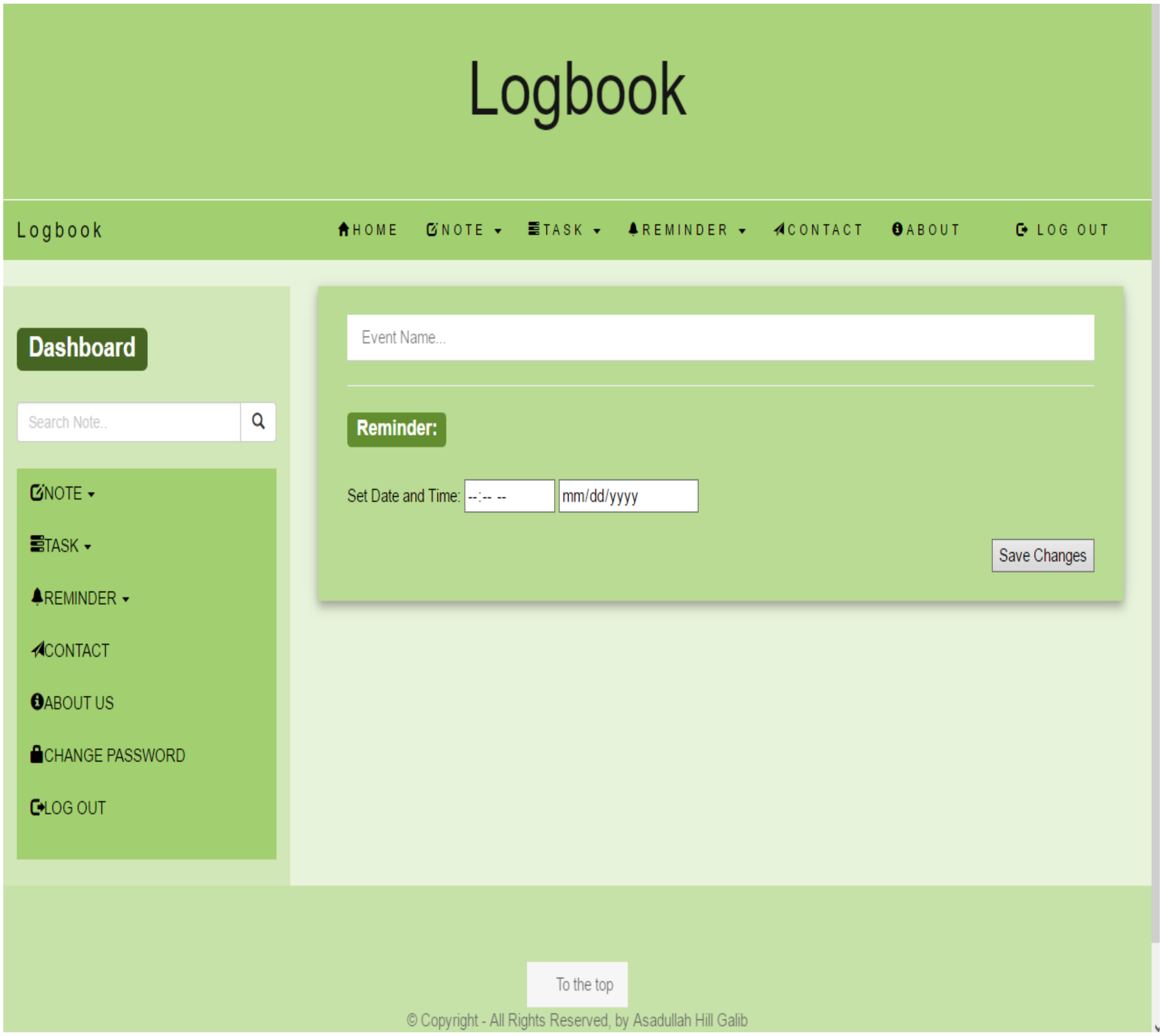
Dealine:

Set Date and Time: 05:00 PM 05/03/2017

Figure 22: User Manual: Adding Task

4.2.5 Adding Reminder Page

After selecting add reminder option, user can set reminder on a particular event. And finally she can save changes.



Logbook

Event Name...

Reminder:

Set Date and Time: --:-- -- mm/dd/yyyy

© Copyright - All Rights Reserved, by Asadullah Hill Galib

Figure 23: User Manual: Adding Reminder

4.2.6 Contact Page

After navigating contact option, user can contact with the server end authority. By this she can complain about anything or give recommendation.

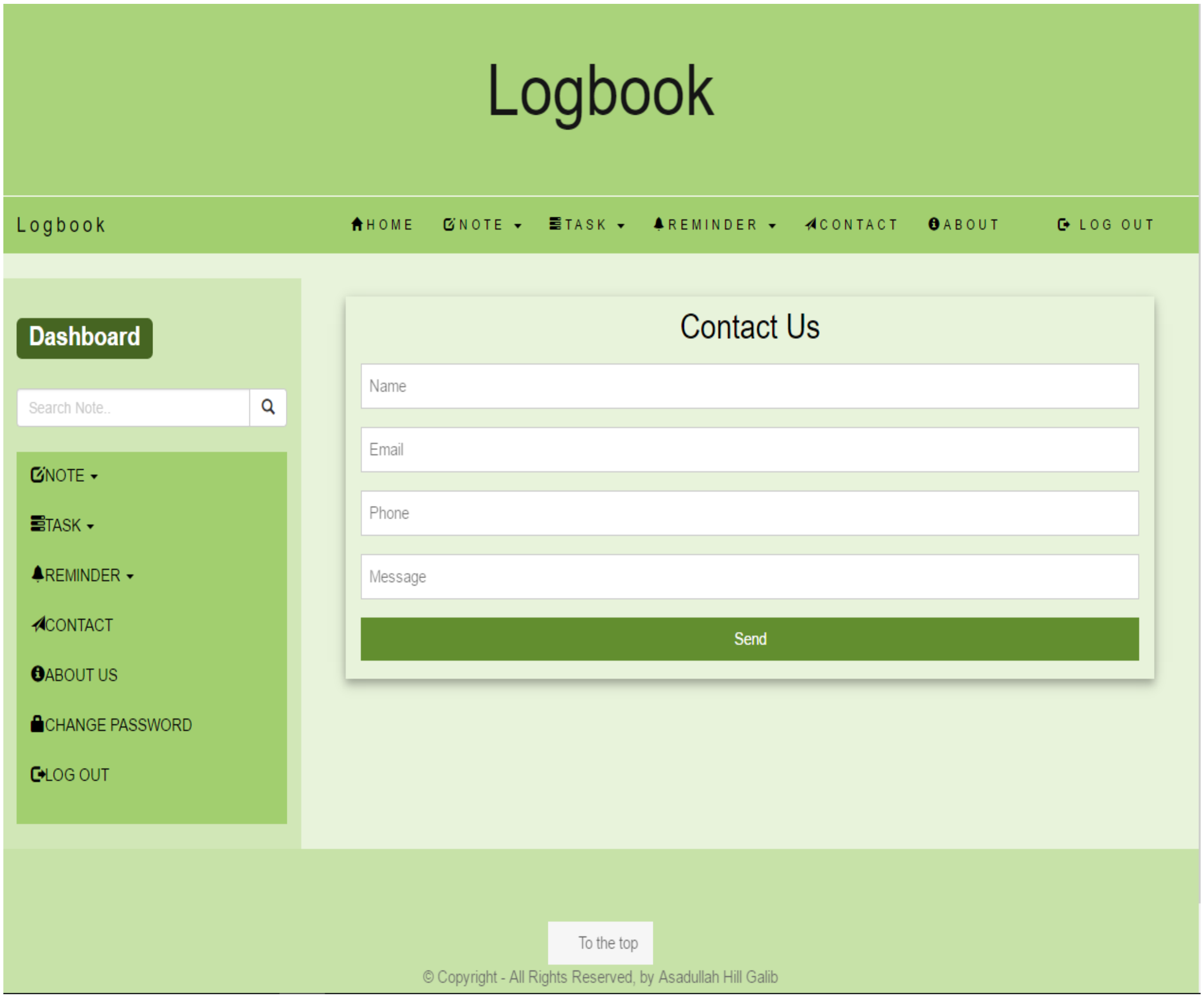
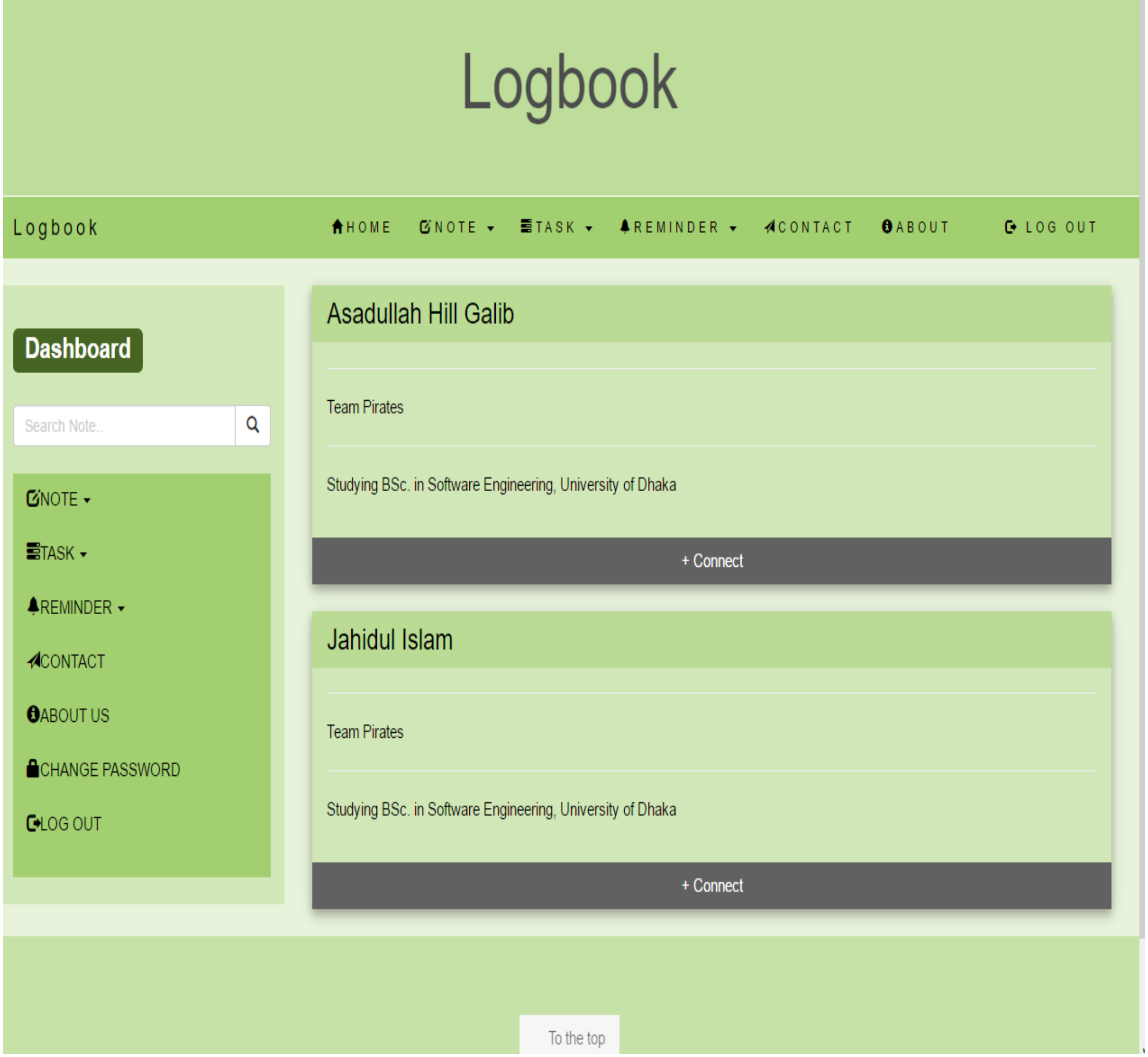


Figure 24: User Manual: Contact

4.2.7 About Us Page

After navigating about us page, user can know about the publisher of this application.



Asadullah Hill Galib

Team Pirates

Studying BSc. in Software Engineering, University of Dhaka

Jahidul Islam

Team Pirates

Studying BSc. in Software Engineering, University of Dhaka

Figure 25: User Manual: About Us

4.2.8 Change Password Page

After user select change password, she can update her account password with a new one for security.

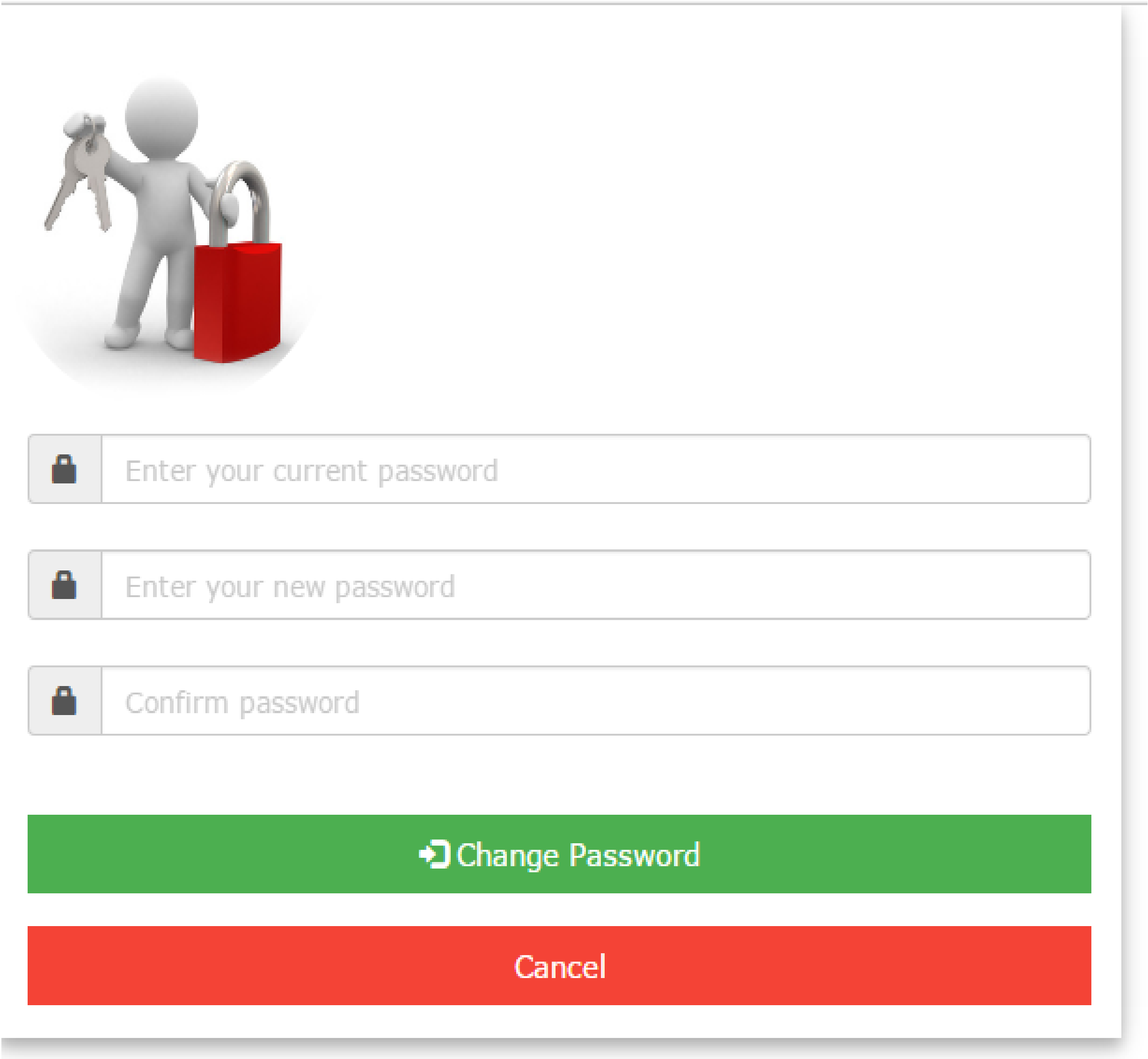


Figure 26: User Manual: Change Password

And User can logged out using log out button.

Chapter 5: Conclusion

Project managers are aware of the difficulty in keeping track of various tasks, resources, timeline in maintaining daily life easily. Our personal task manager is such kind of tool which will be serviceable and easy-to-use. Developing this application was a great experience for us.

5.1 Achievements

This project helped us to achieve a lot of experiences-

- Building first web application
- Creating a real usable software product
- Learning and implementing primitive web technology

5.2 Obstacles

- In spite of using framework, we use raw code. For which we often faces obstacles to handle code.
- To learn and implement web technology simultaneously

5.3 Future Plan

- Updating this application as far as possible
- Creating a android app of this application