

Gulshan Kumar

netBuukkaus – BOOKING SYSTEM

netBuukkaus – BOOKING SYSTEM

Gulshan Kumar
Bachelor's Thesis
Autumn 2015
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology

Author: Gulshan Kumar

Title: netBuukkaus - Booking System

Supervisor: Pertti Heikkilä

Term and year of completion: Autumn 2015

Number of pages: 37

This Bachelor's thesis was done for NetWink Solutions Tmi. NetWink Solutions is a consultancy company which provides search engine optimization services and develops game assets using the blender and Unity3D.

The object of this thesis was to design and develop a stand-alone booking system which includes an admin module and a plugin for a front-end. netBuukkaus is a web application and it was created using PHP and the database was created on the MySQL server. The Front-end plugin allows a user to book the provided services on a specific date and it allows to pay offline. But the user needs a confirmation for a specific booking from the administrator. In the admin module, the administrator is able to add new services, their cost, time, manual booking and a list of pending payment. The administrator can book on behalf of a customer and confirm it. When the administrator confirms the booking, the user will get a confirmation email.

The thesis is divided in two parts. The first part explains, how the user interface works and the second part explains, how database queries and PHP functions work.

At the end of the project, there was a functional booking system. There were also bugs which addressed because of future development. In future, the admin module and front-end plugin will be extended and a new membership module will be made and planned to develop a WordPress plugin.

Keywords: PHP, MySQL, Bootstrap, Modernizr, jQuery

TABLE OF CONTENTS

ABSTRACT	3
TABLE OF CONTENTS	4
ABBREVIATIONS AND TERMS	5
1 INTRODUCTION	6
2 PROJECT STARTING POINT	7
3 FRONT-END PLUGIN	9
3.1 Front-end plugin UI	9
3.2 Plugins Functions	9
4 ADMIN MODULE	11
4.1 Booking Management	13
4.2 Staff Management	15
4.3 Service Management	15
4.4 Payment Management	17
5 TECHNOLOGIES	19
5.1 PHP	19
5.2 MySQL	19
5.3 Bootstrap	19
5.4 jQuery	20
5.5 Modernizr	20
5.6 Wordpress	21
6 FRONT-END PLUGIN IMPLEMENTATION	22
6.1 Retrieve service list	22
6.2 Book a service	23
7 ADMIN MODULE IMPLEMENTATION	24
7.1 Admin and Staff Login	24
7.2 Add, update and display	25
7.2.1 Service	25
7.2.2 Staff	26
7.2.3 Booking	28
7.2.4 Payment	29

8 TESTING	31
9 FURTHER DEVELOPMENT POSSIBILITIES	34
10 CONCLUSION	35
REFERENCES	36

ABBREVIATIONS AND TERMS

Bootstrap front-end framework for web development

ID Identification

URL Uniform Resource Locator



Hide and display pagination icon button



Refresh button to refresh the list



Toggle button



Column button



Edit/update icon button – to edit data from selected row



Display/view icon button – to display data of selected row

1 INTRODUCTION

The netBuukkaus is a completely new stand-alone booking system program that takes advantage of MySQL for storing data and a bootstrap and modernizr framework for developing front-end for users and the administrator. It is suitable for a private tutor, an instructor or a small scale training institute.

This document explains how a front-end plugin for the booking and admin module was carried out. The aim was to make a beta version of a booking system that corresponds to the user's vision.

This thesis is made for NetWink Solutions Tmi. NetWink is consultancy company which provides search engine optimization services and develops game assets in the blender and Unity3D

2 PROJECT STARTING POINT

At the beginning of the thesis front-end UI components were developed. The main development was started by designing a user interface using pre-developed UI components and database designing. The User interface helped to design a table and required functions. After designing the user interface, the main options were developed for a customer, a staff member and the administrator. The Administrator should be able to use all options and they are:

- Service management; create and update services
- Schedule management; create and update a schedule for a selected service
- Staff management; create and update staff
- Booking management; select a service and book a service with required information, a manual booking by the administrator
- Invoice management; update an invoice status such as paid or unpaid, an invoice view
- Service, staff, booking and schedule browsing
- Changing the booking status and payment status

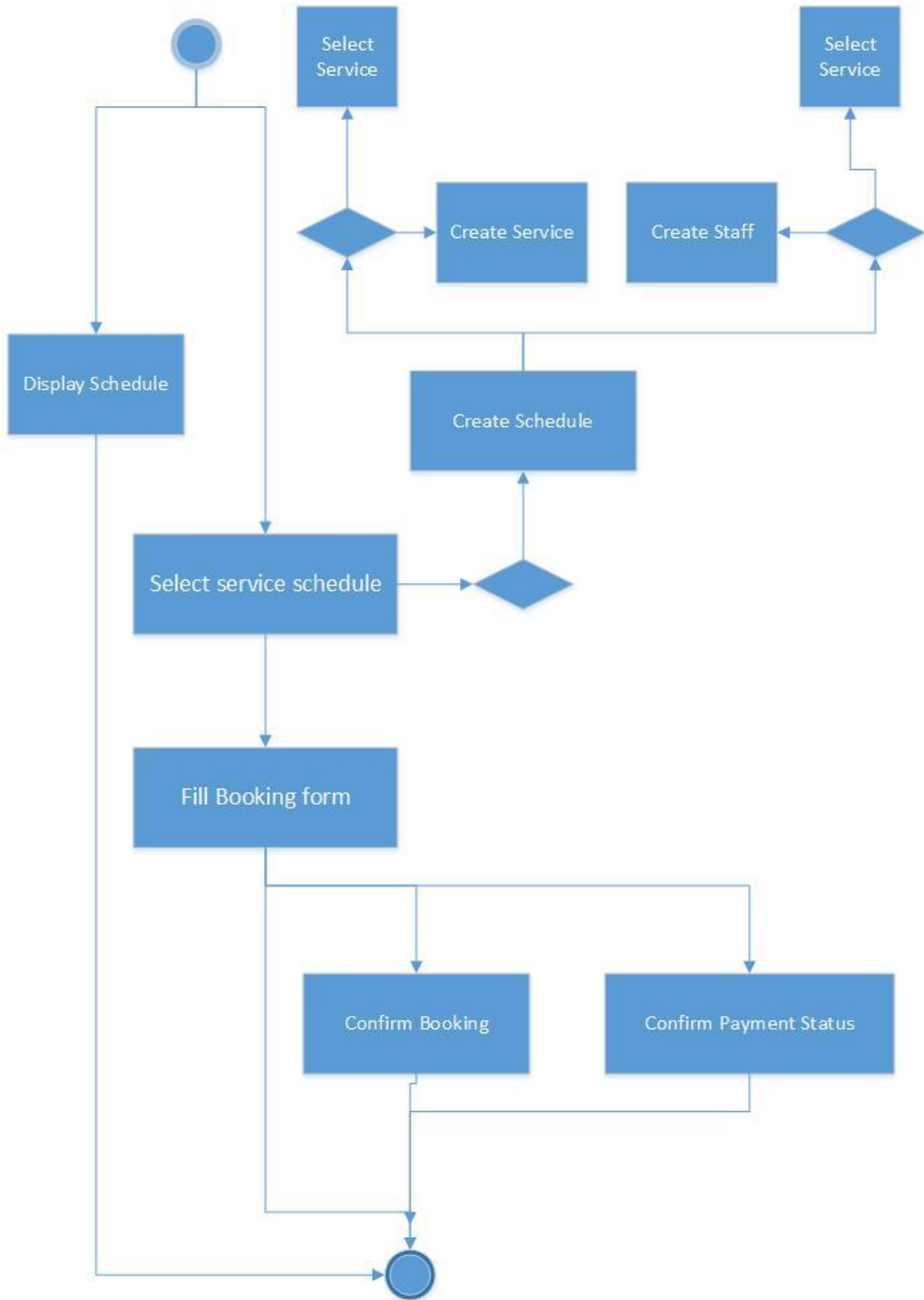


FIGURE 1. Booking system's flow diagram

3 FRONT-END PLUGIN

In this chapter, it is explained how the Front-end booking plugin works so that the ones who are not yet familiar with this plugin can see how it works.

3.1 Front-end plugin UI

When a customer opens the webpage, the main index page will open with currently provided service details shown in figure 2. The user selects the desired service and proceed for booking. For booking, the function requires to submit a name, a phone number, an email, address, and a number of spaces for booking.

3.2 Plugins Functions

The main user interface of the booking system is shown in figure 2. The main interface shows the list of currently available services list. Clicking the 'Book Now' button from the list, the Book service form displays the selected service and date in a disabled mode. Then you enter all required data. If the submit button is pressed, without the required information, the required field displays a notification as shown in figure 3.

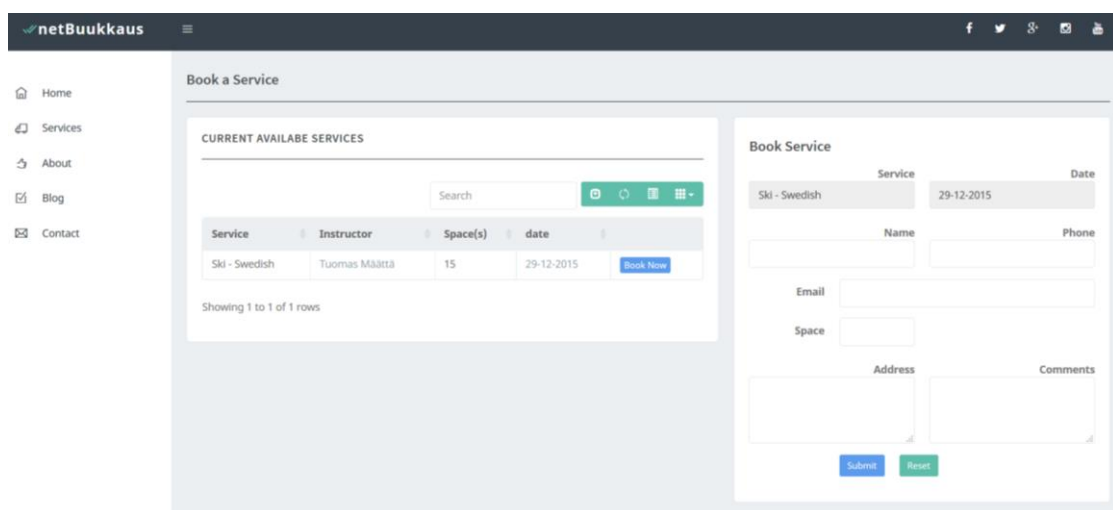


FIGURE 2. Booking Plugin – the main user interface

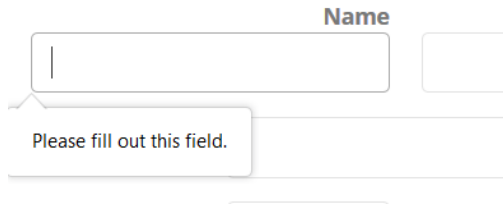


FIGURE 3. Required input field notification

After entering all required information and pressing the submit button, the user will get successfully a data submission message shown in Figure 4. Now the booking request requires a confirmation by the administrator. The user will get a booking confirmation when the administrator accepts the booking.

Book Service

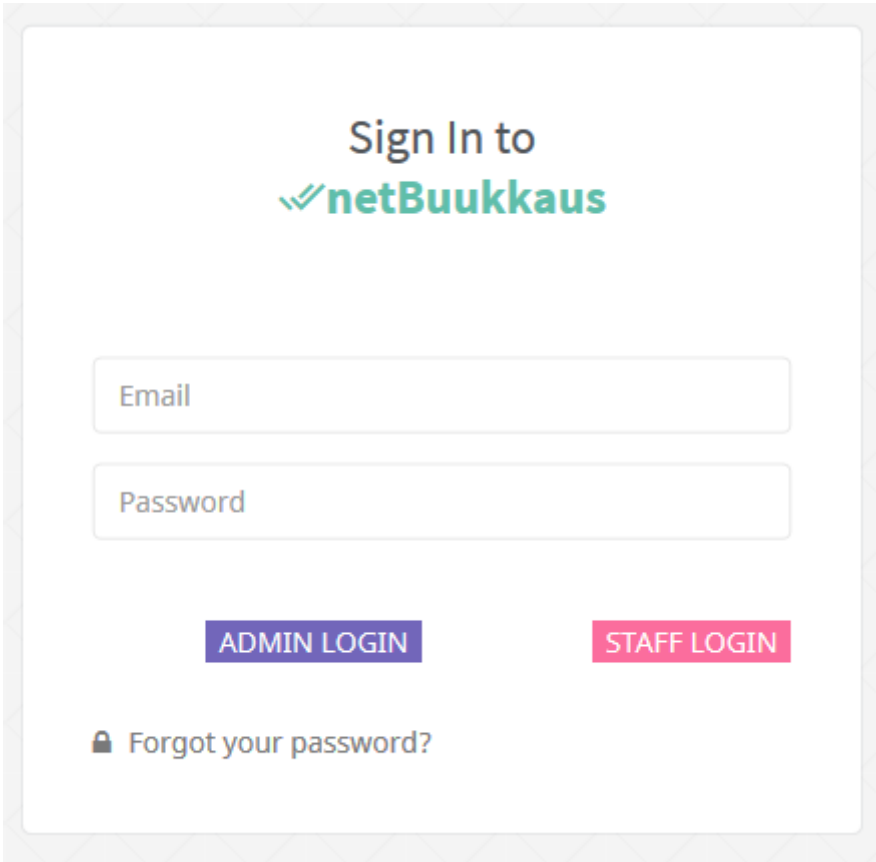
You will get confirmation soon

FIGURE 4. Confirmation message

4 ADMIN MODULE

In this chapter the user interface for the admin module is explained so that the reader will get familiar with this booking system and see how it works.

The administrator and staff login user interface is shown in Figure 5. The administrator and staff member can login with their given email and password by pressing the button according to their access right, such as admin or staff.



The image shows a login interface for 'netBuukkaus'. At the top, it says 'Sign In to netBuukkaus'. Below this are two input fields: 'Email' and 'Password'. At the bottom, there are two buttons: 'ADMIN LOGIN' (purple) and 'STAFF LOGIN' (pink). A link 'Forgot your password?' with a lock icon is also visible.

FIGURE 5. Administrator and Staff Login Screen

After the administrator has logged in, a dashboard interface appears as shown in Figure 6. From the dashboard (new services, new events/schedule, manual booking) a staff member can create and update the user's payment status and view a list of all services. The dashboard screen displays today's schedule, a pending payment, staff, a list of current services and a menu for different options.

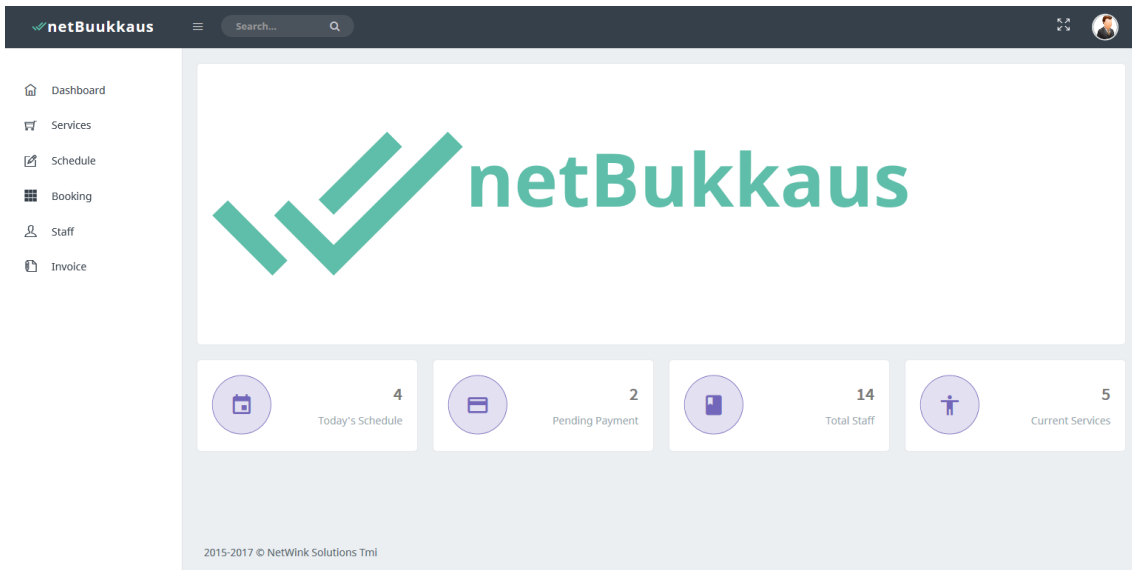


FIGURE 6. Administrator Dashboard

When a staff member has logged in, the staff dashboard screen will appear. It shows the booking schedule, which includes service and attendee details, such as a booked space, date and their status. When the service name is clicked, it shows the service details and the name of the attendee as shown in Figure 7. In booking, the schedule list status displays the booking status confirmed or pending in different colours, which helps the staff member to recognize and search the data after entering a value in search the input box.

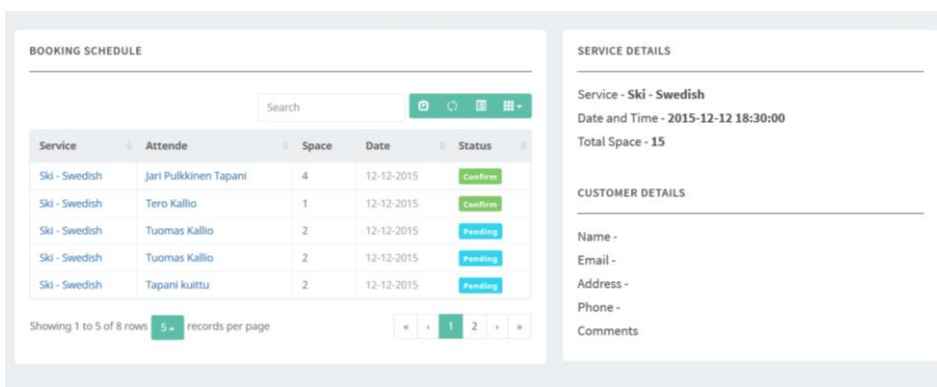


FIGURE 7. Staff Dashboard

All lists in the booking plugin for the front-end and admin module contain a button group which contain toggle, a pagination, a column and a search input box shown in the top of the table as shown in figure 8 to search and display the data in table.

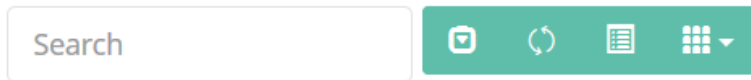


FIGURE 8. Search, hide pagination, refresh, toggle and column buttons

4.1 Booking Management

Adding a new booking is done by pressing Add Booking buttons and updating the existing booking is done by pressing a pencil icon button as shown in Figure 9. The Booking list shows the booking status in different colours based on their status. The list shows the customer's name, the selected service and the instructor assigned to a specific service, a date of service/event, spaces booked by the customer and their status.

Booking

Add Booking

Search

Customer Name	Service	Instructor	Date	quantity	Status	Action
Jari Pulkkinen Tapani	Ski - Swedish	Tuomas Määttä	29-12-2015	4	Pending	
Tero Kallio	Ski - Swedish	Tuomas Määttä	29-12-2015	1	Confirmed	
Tuomas Kallio	Ski - Swedish	Tuomas Määttä	29-12-2015	2	Pending	
Tuomas Kallio	Ski - Swedish	Tuomas Määttä	29-12-2015	2	Pending	
Tapani kuittu	Ski - Swedish	Tuomas Määttä	29-12-2015	2	Pending	

Showing 1 to 5 of 9 rows 5 records per page

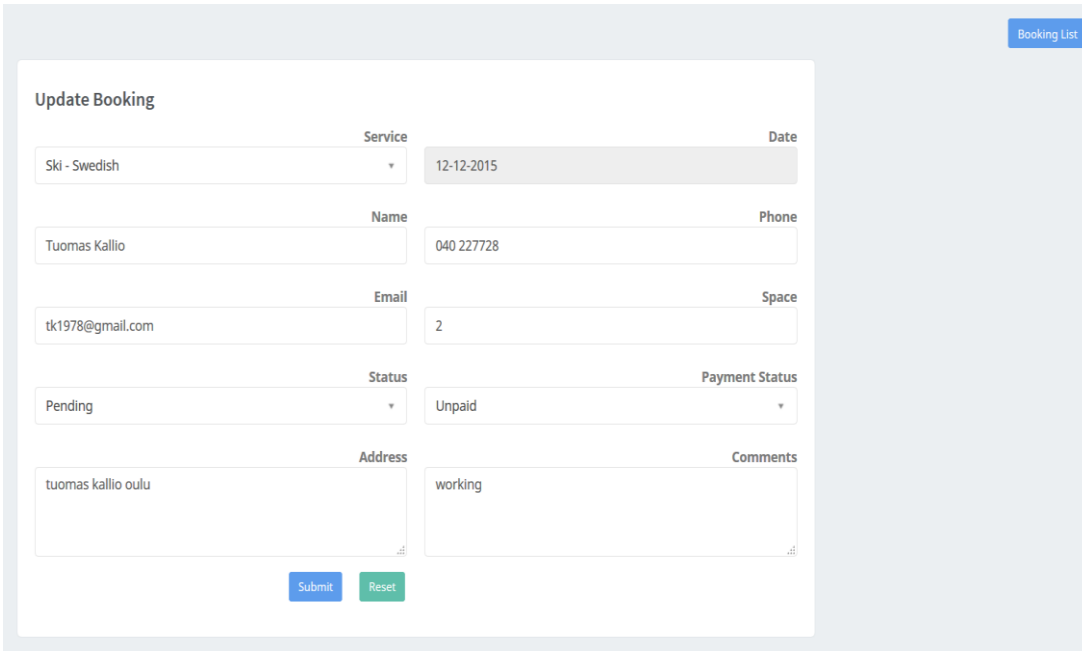
« < 1 2 > »

FIGURE 9. Booking list

All data fields are compulsory when booking a service. After filling in the data, a submit button is pressed. It saves the data and displays a confirmation mes-

sage on the screen after a successful insertion. The Booking plugin is the same as in the customer booking plugin. The Administrator can manually book the service on behalf of the customer and confirm the service.

To update a booking, you need to press the pencil icon button. It opens a new page for updating with a selected booking id and displays booking details in forms as shown in Figure 10. To modify data, you need to press a submit button to update the selected booking. After a successful submission, a message appears on the screen and it updates the booking list. The booking list can be checked by pressing the Booking List button or from menu the left- bar menu. A Date cannot be changed by the administrator while updating a booking except service details. The date will update when the service schedule is changed and submitted. The booking update can be seen in the booking list as shown in Figure 9.



The screenshot shows a web form titled "Update Booking" within an administrator dashboard. The form is organized into several rows of input fields. The first row contains a dropdown menu for "Service" (selected as "Ski - Swedish") and a date field for "Date" (12-12-2015). The second row has a text input for "Name" (Tuomas Kallio) and a text input for "Phone" (040 227728). The third row includes a text input for "Email" (tk1978@gmail.com) and a text input for "Space" (2). The fourth row features a dropdown for "Status" (Pending) and a dropdown for "Payment Status" (Unpaid). The fifth row consists of a text input for "Address" (tuomas kallio oulu) and a text area for "Comments" (working). At the bottom of the form are two buttons: "Submit" (blue) and "Reset" (green). In the top right corner of the dashboard area, there is a blue button labeled "Booking List".

FIGURE 10. Edit/update booking – administrator dashboard

During the booking update, the booking status and payment status can be changed too. The payment status can also be changed while viewing the invoice list.

4.2 Staff Management

The Staff list displays a staff member's details either active or disabled status with their position. Currently, staff means instructors. To enter a new staff member you need to fill in the required details and press the submit button. Name, email and password details are compulsory for a staff member to login. After successful submission, a message appears on the screen. In case of editing existing staff, you need to press the pencil icon button which displays the details in an empty field in the form. Then you need to update the information and press the submit button which updates the information and the list is shown in Figure 11. To search a specific email or position or name, you enter a text in the search box and the list will display the search result.

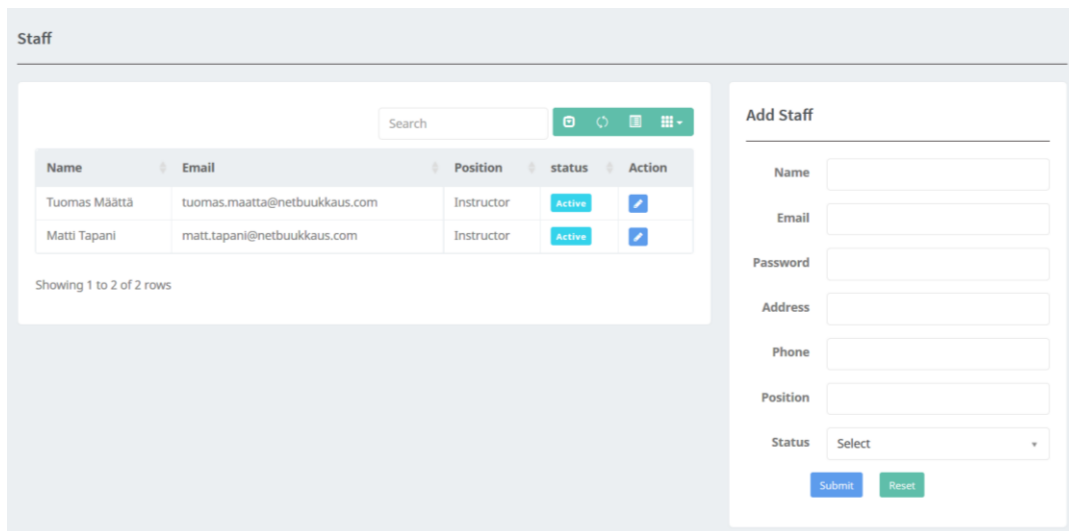


FIGURE 11. Staff list

4.3 Service Management

Like the staff list, the service list displays the existing services, either active or disabled status services list, as shown in Figure 12. The Administrator adds new services by pressing the submit button and modifies the existing services by pressing the pencil icon button.

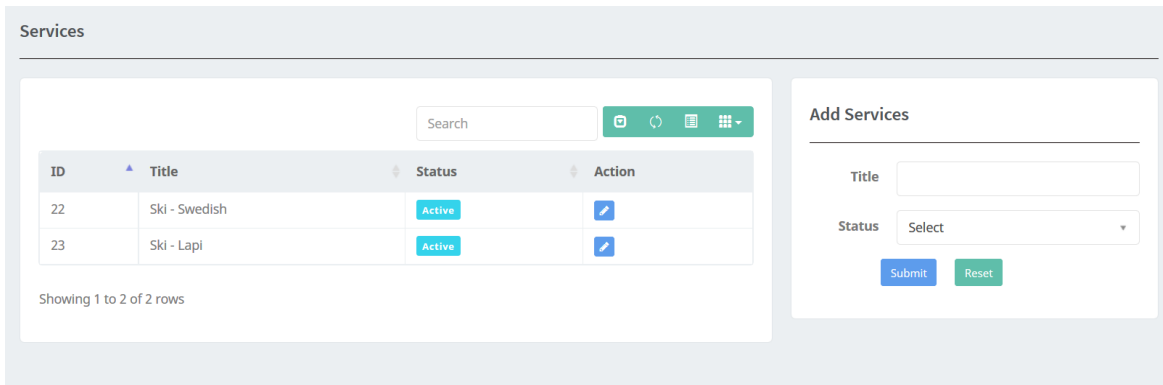


FIGURE 12. Service List

The Service schedule requires that there is at least one service where the schedule can be added. The Service schedule includes the service name, date, time, instructor and allowed space. The Service schedule data is added to the booking and service schedule table. To add a new schedule, you need to press the add schedule button or select from the menu and to edit existing schedule, you need to press the pencil icon button from the list. It gets the selected service schedule id and opens on a different web page, where the administrator modifies the service schedule and submits it. After the submission, a confirmation message appears on the screen. If any error occurs, a connection error message appears on the screen.

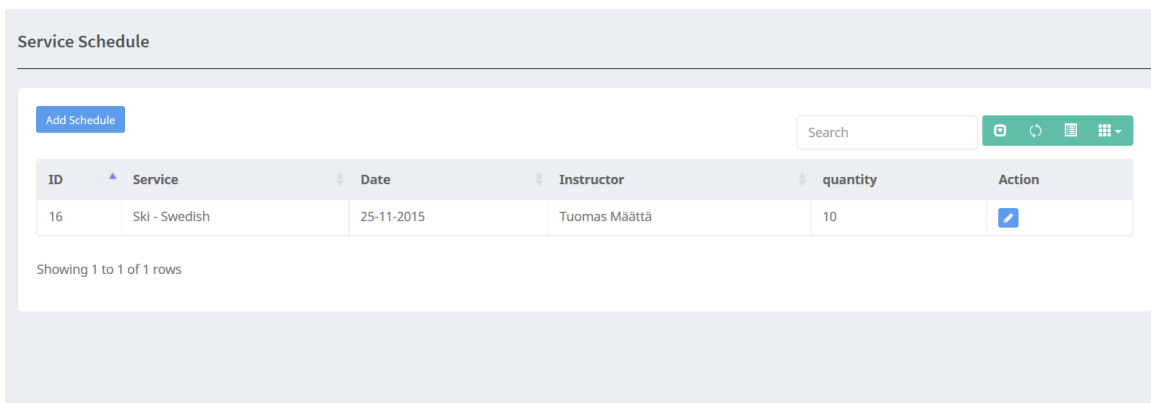


FIGURE 13. Service schedule list

To create a service schedule, you need to select the service title and instructor name. All data is compulsory to create a schedule, except comments. The Comments are commonly used here to provide some instructions regarding service, location or precaution. The Service schedule is updated in the same way as staff, booking and service. Selecting the pencil icon button from the list opens the schedule, updates the data when pressing the submit button, the confirmation will appear on the screen (see Figure 14).

The screenshot shows a web form titled "Service Schedule" with a sub-header "Add Schedule". The form contains the following fields and controls:

- Services:** A dropdown menu with "Ski - Swedish" selected.
- Instructor:** A dropdown menu with "Tuomas Määttä" selected.
- Available Spaces:** A text input field containing "15".
- Start Date:** A date picker showing "12-29-2015" with a calendar icon.
- Start At:** A time picker showing "18:30:00" with a clock icon.
- Duration (in min):** A text input field containing "120".
- Price (€):** A text input field containing "150".
- End Date:** A date picker showing "12-29-2015" with a calendar icon.
- End At:** A time picker showing "20:45:00" with a clock icon.
- Comments:** A large empty text area.
- Buttons:** "Submit" (blue), "Reset" (green), and "Schedule List" (blue, top right).

FIGURE 14. Adding and updating service schedule

4.4 Payment Management

The Payment List is shown in Figure 15. The Administrator can see the status of a booking payment either unpaid or paid. If the user has paid the booking payment, the administrator can update it by pressing the pencil icon button to select the booking and change the payment status as other input boxes are disabled.

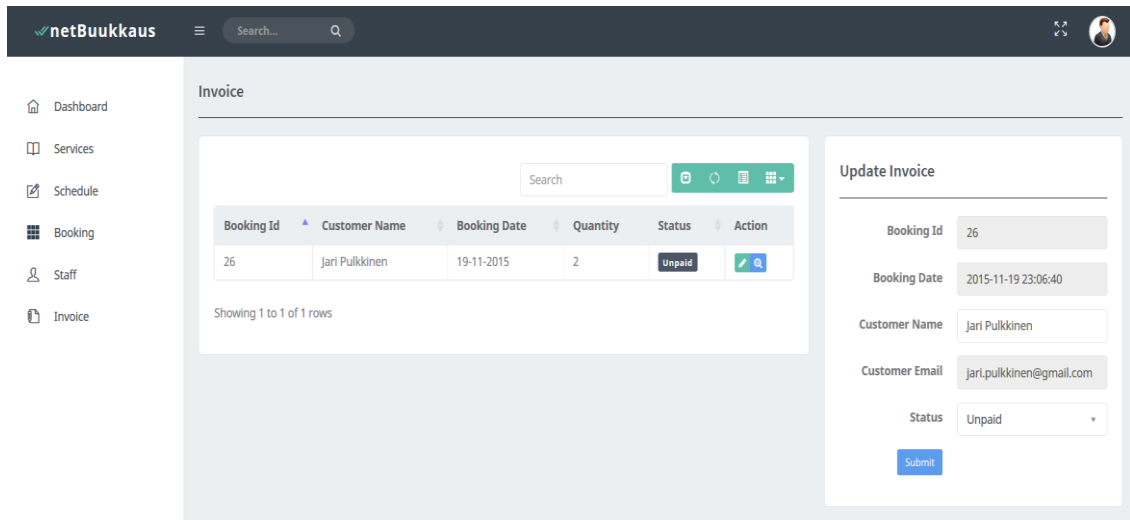


FIGURE 15. Invoice List with status

By pressing the search-plus button, the administrator can view the selected invoice in the browser and by pressing the Invoice List button, it displays the invoice list.

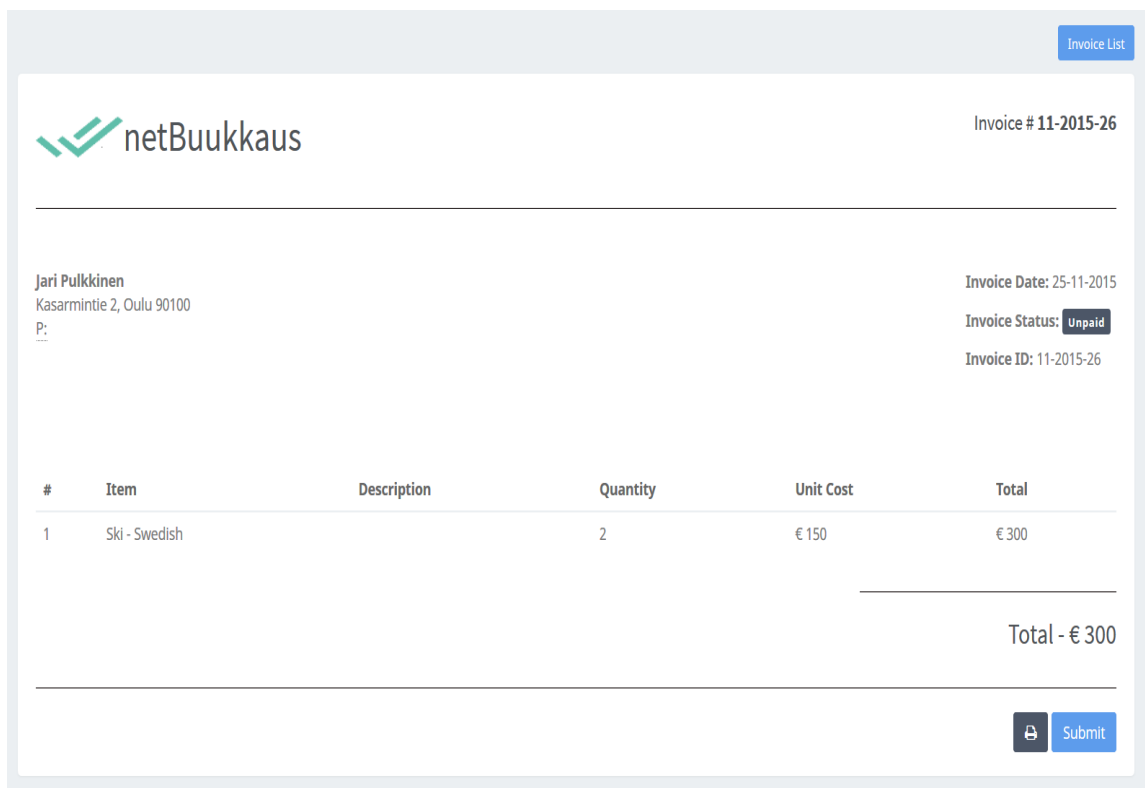


FIGURE 16. Detailed invoice

5 TECHNOLOGIES

5.1 PHP

PHP is an acronym for “PHP: Hypertext preprocessor” and widely used, open source scripting language. It is mainly focused on server-side scripting and it is associated with a web-based application.

```
function count_services(){  
    global $datab;  
    $query = "SELECT COUNT(*) FROM services WHERE status=1";  
    $result = mysqli_query($datab, $query);  
    $count = mysqli_fetch_array($result);  
    echo $count[0];  
}
```

FIGURE 27. Sample PHP function query

5.2 MySQL

MySQL is an open source Relational Database management system that runs on a server for both a small and large application and it uses the standard SQL.

```
$query = "SELECT COUNT(*) FROM booking WHERE id=".$ser_sch_id;
```

FIGURE 38. Sample MySQL query

5.3 Bootstrap

Bootstrap is a front-end framework that helps developers to jump start the web development process. (1.) In general, Bootstrap contains CSS and HTML based design templates for navigations, a button, fonts and other components as well as JavaScript extensions.

```
<div class="col-md-5">  
    <input class="pull-left btn btn-sm btn-default" id="create" name="create" value="Reset" type="reset">  
</div>  
..
```

FIGURE 49. Sample bootstrap class code

5.4 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things, like an HTML document traversal, and manipulation, event handling, animation and Ajax is much simpler with an easy-to-use API that works across a multitude of browsers. (10)

```
$('#input#aloptions').maxlength({
  alwaysShow: true,
  warningClass: "label label-success",
  limitReachedClass: "label label-danger",
  separator: ' out of ',
  preText: 'You typed ',
  postText: ' chars available.',
  validate: true
});
```

FIGURE 20. Sample jQuery code

5.5 Modernizr

Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user's browser. It tells if the browser is visiting a website, what features that browser supports and what features it does not support. It makes easy to write conditional JavaScript and CSS to handle each situation so that the browser either supports a feature or not.

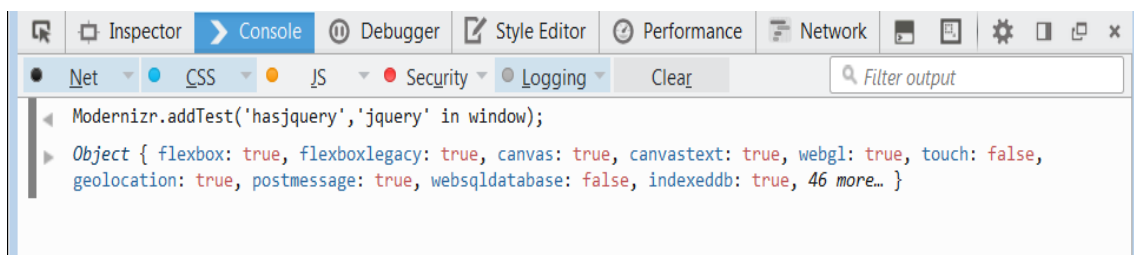


FIGURE 21. Modernizr addTest API detecting jquery in browser console

5.6 Wordpress

WordPress is most popular blogging and content management system (CMS) based on PHP and MySQL. Some use WordPress as a tool for creating and managing any kind of website. It is used worldwide by many companies for blogs and fully featured websites. WordPress is free and possible to download from wordpress.org. WordPress is open source which means that all the files and codes are available for customizing and enhancing. Its main feature are thousands of templates, plugin (see Figure 22) and search engine friendly link building. For business WordPress is a versatile technology.

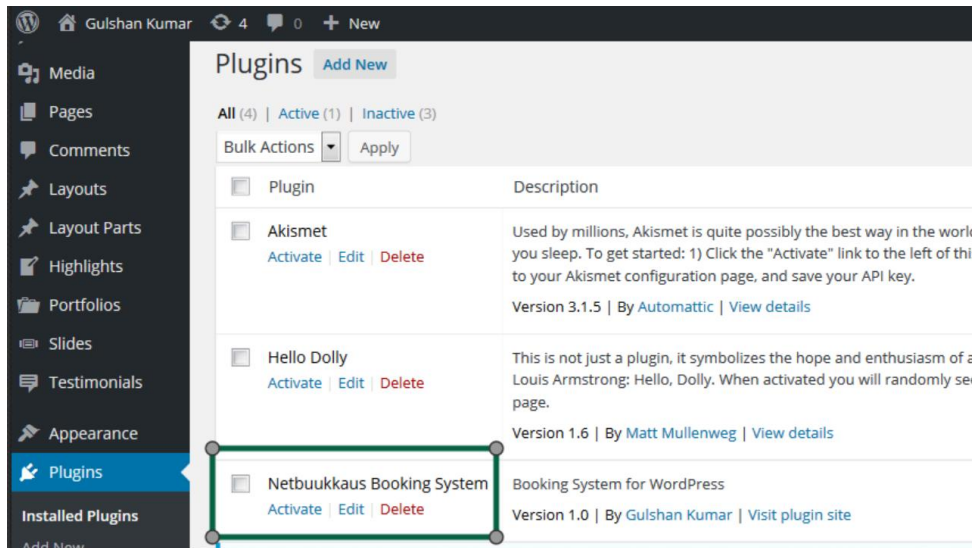


FIGURE 22. WordPress netBuukkaus booking system's demo plugin

6 FRONT-END PLUGIN IMPLEMENTATION

This chapter explains how the booking plugin was implemented and describes its functions in more detail. The Booking plugin was carried out using server scripting language PHP and MySQL.

The billing plugin data is stored in one main MySQL table named as booking. But it retrieves data from three different tables - staff, services, service schedule. The booking table is intended to store the booking id, service schedule id, customer name, customer address, customer phone number, customer email, customer address, comments, and number of spaces, booking confirmation status and booking payment status. The Booking table is connected to the service schedule table to get service details such as time, available spaces, and staff id. The Service schedule table is connected to two different tables. First, services to store service id and title and staff to store staff id, name, designation, email, password and address.

6.1 Retrieve service list

While designing the booking plugin, the main problem was to develop a function using MySQL query to retrieve a service list with schedule and staff details. To get the required result, inner join command used to join three tables to get service name, date, available space and instructor details. Then the results of query was assigned to table's column. To retrieve the date from the datetime column, a date_format command was used.

```
$query = "select booking.id, booking.customer_name, booking.qty, booking.status,
services.title, staff.name, (booking.qty*service_schedule.price) as amount,
date_format(service_schedule.date_time_start,'%d-%m-%Y') as date_time_start from booking
inner join service_schedule on booking.ser_sch_id = service_schedule.id inner join
services on service_schedule.id = services.id inner join staff on
service_schedule.instructor = staff.id " ;
$result = mysqli_query($datab,$query) or die("Oh crap...: " . mysql_error());
```

FIGURE 23. Retrieve service list with schedule and staff details

6.2 Book a service

In the front-end booking plugin, the function displays only the current services with their schedule, instructor, date and available spaces with a book now button. The Book now button gets the service schedule id with the service name and date and waits for the customer to enter the required details. It uses the query shown in figure 23. The Booking id is a primary key and auto generated. The Booking status default is 2 which stands for pending and the payment status is default 2 which stands for unpaid.

If the booking form is not filled correctly or any input field is empty, the function shows a notification. After filling in the data correctly, the submit button is pressed to book the desired service. The Function checks the customer name, customer email and service schedule. If they are empty, then the function displays an alert message. Otherwise, data is inserted in the booking table as shown in figure 24.

```
if (isset($_POST ['book'])) )
{
    $sid= $_POST['service_s_id'];

    if (!empty($customer_name) && !empty($customer_email) && !empty($service_s_id) ) {
        $q = "INSERT INTO booking
        (ser_sch_id, customer_name, customer_email, customer_phone, customer_address, customer_comments, status, p_s
        tatus, created_on)
        VALUES (" . $service_s_id . ", " . $booked_space . ", " . $customer_name . ", " . $customer_email .
        ", " . $customer_phone . ", " . $customer_address . ", " . $customer_comments . ", 2, 2, NOW() )";
        $res = mysqli_query($datab, $q) or die("error!");
        header('Location: '.$_SERVER['PHP_SELF'].'?created=You will get confirmation soon');
        exit;
    } else
    {
        echo "<script type= 'text/javascript'>alert('Select service');</script>";
    }
}
```

FIGURE 24. Inserting booking details

7 ADMIN MODULE IMPLEMENTATION

This chapter explains how the admin module was implemented and describes its functions in more detail. The Admin module was carried out using a server scripting language PHP.

7.1 Admin and Staff Login

In the admin module, the first program was implemented by creating a login system for admin and staff. Data is stored in two separate MySQL tables. The first table is an admin table and it is intended to store admin details such as name, email, password, and avatar. The second table is staff which contains the same information as the admin table. Both a correct email and a password are required to enter the dashboard (see Figure 25).

```
<div class="form-group ">
  <div class="col-xx-12">
    <input class="form-control" type="email" required="" id="email" name=
      "email" placeholder="Email">
  </div>
</div>

<div class="form-group">
  <div class="col-xx-12">
    <input class="form-control" type="password" required="" id="password"
      name="password" placeholder="Password">
  </div>
</div>
```

FIGURE 25. Admin and staff login

After entering the correct email and password, the function checks if the provided information exists in table, if yes, the dashboard appears. Otherwise, it will show the incorrect login details message and return to the login page. The login checking function is shown in figure 26. The function uses the both table admin and staff to check login details.

```

$email = mysqli_real_escape_string($datab,$_POST['email']);
$password = mysqli_real_escape_string($datab,$_POST['password']);
$select_user = "select * from admin where email='$email' AND password='$password'";
$run_user = mysqli_query($datab, $select_user);
$num_rows = mysqli_num_rows($run_user);
if($num_rows==1){
    $_SESSION['email']=$email;
    echo "<script>window.open('dashboard.php?logged_in=You have successfully Logged in!','_self')</script>";
}

```

FIGURE 26. Admin and staff logging in

7.2 Add, update and display

Adding, updating and viewing services, the service schedule, booking and viewing invoices list are done by the administrator. A new administrator is required to be created manually as adding a new administrator function which does not exist yet. The administrator is the only one who can add new staff members.

7.2.1 Service

The service list and adding a new service function is implemented after developing the admin and staff login function. The function add services and data is added to service table. The service status is default 1, which stand for active. The Administrator can update it to 2. Before entering the service, the function check if the service name is empty or not. If the service name is empty, then it displays an error message, otherwise it inserts a new service.

The Update function checks all the services before it updates the selected service. The Function gets id from the selected service as shown in figure 27 and displays data in the service form. After modifying the data function, the service name is checked. If the service name is not empty, it updates the data, otherwise it displays the error message.

```

if (isset($_GET['id']) && is_numeric($_GET['id']) && $_GET['id'] > 0)
{
    $id = $_GET['id'];

    $result = mysqli_query($datab,"SELECT * FROM services WHERE id=$id") or die(mysqli_error
    ());
    $row = mysqli_fetch_array($result);
    if($row)
    {
        $service_title = $row['title'];
        $service_id = $row['id'];
    }
    else
    {
        echo "No results!";
    }
}

```

FIGURE 27. Getting service id and assigning value

Every table list contains a button group with a search input box shown in figure 28. When any alphabet or digit is entered in search box, the function searches the data from the list and updates the list according to the entered data in the search box. The Function searches data based on the column, hides the pagination, and refreshes the list and viewing list.

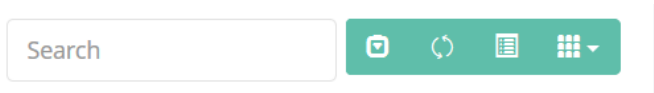


FIGURE 28. Search input box with button group

7.2.2 Staff

The Staff list and adding a new staff member function are implemented after developing the service list and adding new services. Before inserting a new staff member, the function checks if the staff name is not empty then it inserts data, otherwise the function displays connection error message windows using JavaScript. All the fields are required to insert staff details.

```

if (!empty($name) ) {
    $q = "INSERT INTO staff (name,email,password,address,
    phone,position,status,created_on)
    VALUES (' . $name . ',' . $email . ',' . $password . ',' . $address .
    ',' . $phone . ',' . $position . ',' . $status . ',' . NOW() )";
    $res = mysqli_query($datab, $q) or die("error!");
    header('Location: '.$_SERVER['PHP_SELF'].'?created=New Staff Member Created');
    exit;
}

```

FIGURE 29. Insert staff member

The Function updates staff details after selecting the staff from the list by clicking the pencil icon button and gets the staff id. Then the function searches the staff details where id is equal to the selected id from the list. If it is found, then the function displays the details on the form.

```

$result = mysqli_query($datab,"SELECT * FROM staff WHERE id=$id") or die(
mysqli_error());
$row = mysqli_fetch_array($result);
if($row)
{
    $staff_id = $row['id'];
    $staff_name = $row['name'];
    $staff_email = $row['email'];
    $staff_password = $row['password'];
    $staff_address = $row['address'];
    $staff_phone = $row['phone'];
    $staff_position = $row['position'];
    $staff_status = $row['status'];
}

```

FIGURE 30. Search staff by id

After updating the staff information, the submit button is pressed to update. After pressing the submit button, the function checks if the submit button id is update then it checks if the staff name is not empty, as when inserting. If the submit button is pressed and the name is not empty, then the function updates the data and displays a message on screen.

```

if (isset($_POST ['update'])) )
{
    $id= $_POST['id'];

    if (!empty($name) ) {

        $query = mysqli_query($datab,"UPDATE staff SET name ='".$name."', email ='".$email.
        "', password ='".$password."', address ='".$address."', phone ='".$phone."', position
        ='".$position."' WHERE id ='".$id);

        if ($query) {
            header('Location: '.$_SERVER['PHP_SELF'].'?updated=Staff Data Updated');
            exit;
        } else
        {
            echo "<script type= 'text/javascript'>alert('connection error');</script>";
        }
    }
}

```

FIGURE 31. Update staff details

7.2.3 Booking

The Function gets all booking details after joining all four tables – booking, service schedule, staff and service, and displays them in the booking list. To update, the booking function gets an id and searches the necessary information based on the id. If the id exists in the booking table, then it gathers all necessary data from all other tables and displays them in the table list.

```

$id = $_GET['id'];

$result = mysqli_query($datab,"select booking.*, services.title,
services.id as sid, service_schedule.id as ssid, staff.id as stid,
date_format(service_schedule.date_time_start,'%d-%m-%Y') as
date_time_start from booking inner join service_schedule on
booking.ser_sch_id = service_schedule.id inner join staff on
service_schedule.staff_id = staff.id inner join services on
services.id= service_schedule.service_id where booking.id =$id" ) or
die(mysqli_error());
$row = mysqli_fetch_array($result);

```

FIGURE 32. Update booking

The Booking list table shows by default 5 bookings on every page and it can be changed by the user to 10 and 20. But if data-page-size="5" is change to 10, then it displays 10 records in the list. The search input box can be removed if data-search="true" is changed to false then the search input box would not appear in the button toggle list.

```

<table data-toggle="table"
  data-search="true"
  data-show-refresh="true"
  data-show-toggle="true"
  data-show-columns="true"
  data-sort-name="id"
  data-page-list="[5, 10, 20]"
  data-page-size="5"
  data-pagination="true" data-show-pagination-switch="true" class="table-bordered ">
  <thead>
    <tr>
      <th data-field="id" data-sortable="true" data-formatter="">Booking Id</th>
      <th data-field="customername" data-sortable="true">Customer Name</th>
      <th data-field="bookingdate" data-sortable="true">Booking Date</th>
      <th data-field="quantity" >Quantity</th>
      <th data-field="status" data-sortable="true" >Status</th>
      <th data-field="action">Action</th>
    </tr>
  </thead>
  <tbody>
    <?php get_all_invoice(); ?>
  </tbody>
</table>

```

FIGURE 33. Display booking list

7.2.4 Payment

All the booking data is stored in the booking table which includes the payment status too. In the program, 1 is used for paid and 2 is used for unpaid. In the beginning, 0 was planned to use for unpaid payment status but function was returning error message when inserting 0. So 2 is used for an unpaid and then the program works well and it returned a desired result.

```

<div class="form-group">
  <label class="col-md-5 control-label">Status</label>
  <div class="col-md-7">
    <select id="payment_status" name="payment_status" class="form-control select2">
      <option>Select</option>
      <option <?php if ($booking_payment_status == 1) { echo 'selected'; };?>
        name="1" value="1">Paid</option>
      <option <?php if ($booking_payment_status == 2) { echo 'selected'; };?>
        name="2" value="2">Unpaid</option>
    </select>
  </div>
</div>

```

FIGURE 34. Update and display payment status

Rather than creating a new button for update, simple if statement is used to change the button id from create to update if form get an id.

```
<div class="form-group">
  <div class="col-md-5">
    <input class="pull-right btn btn-sm btn-primary" type="submit" id="<?php if (
      isset($_GET['id'])){ echo "update";} ?>" name="<?php if (isset($_GET['id'])){
        echo "update";} e ?>" value="Submit">
    </div>
  </div>
</div>
```

FIGURE 35. Detecting submit button pressing

The Function shown in figure 36 changes the label colour. These labels are in list for status and payment status too.

```
function statusFormatter(value, row) {
  var labelColor;
  if (value == "Paid") {
    labelColor = "success";
  }else if(value == "Unpaid"){
    labelColor = "warning";
  }else if(value == "Pending"){
    labelColor = "info";
  }else if(value == "Cancel"){
    labelColor = "danger";
  }
  var icon = row.id % 2 === 0 ? 'fa-star' : 'fa-user';
  return '<div class="label label-table label-'+ labelColor+'"> ' + value + '</div>';
}
```

FIGURE 36. Status formatter function for table list

8 TESTING

The Test plan was that I would test the function to check if the program is working as expected and MySQL tables query. The Booking system testing was also done by a random tester to check the functionality. netBuukkaus was temporarily installed on netbuukkaus.atwebpages.com for the testing purpose. The Testing was done using w3c develop tools. After testing in W3C's unified validator, it showed seven errors as shown in figure 37.

URI: http://netbuukkaus.atwebpages.com/			
134	36	<th data-field="spaces" data-sortable="true" data-sorter="priceSorte...	No space between attributes.
168	52	<h4 style="color:gray; text-align:center;"></h4 >	End tag h5 seen, but there were open elements.
169	53	<h4 style="color:Green; text-align:center;"></h4 >	End tag h5 seen, but there were open elements.
214	77	...tarea class="form-control" id="address" name="address" rows="2" " " ></textarea>	Saw " when expecting an attribute name. Probable cause: = missing immediately before.
214	80	...tarea class="form-control" id="address" name="address" rows="2" " " ></textarea>	Attribute " not allowed on element textarea at this point.
218	79	...rea class="form-control" id="comments" name="comments" rows="2" " " ></textarea>	Saw " when expecting an attribute name. Probable cause: = missing immediately before.
218	82	...rea class="form-control" id="comments" name="comments" rows="2" " " ></textarea>	Attribute " not allowed on element textarea at this point.

FIGURE 37. Unicorn – W3C's unified validator

Webpagetest.org was used to check the web page performance, because webpagetest is used to analyse and measure the performance of a web page.

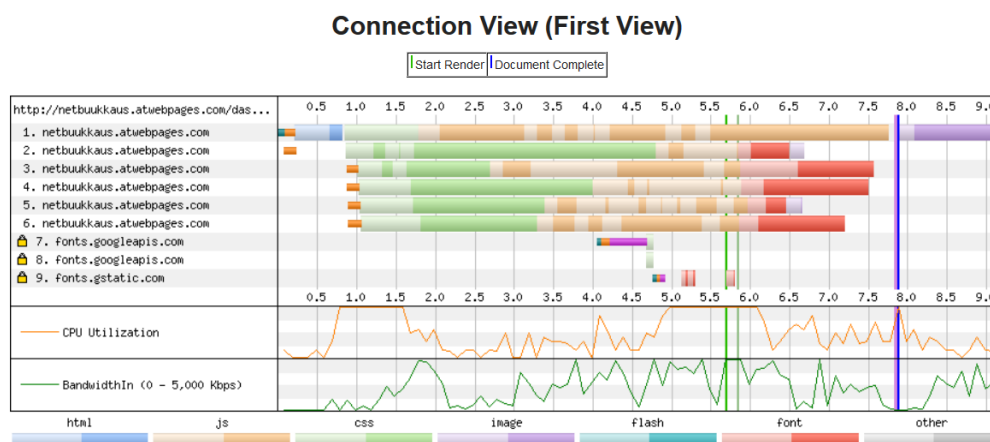


FIGURE 38. Web Page Performance Test for dashboard.php

As a test result, a number of programming errors or bugs were found specially when the test was conducted by a random user. Another user or tester thinks differently and creates a new situation with new question that programmer does not realized. After testing the program, valuable test results were gained, how the program works on other machines, operating system and different browser. The test result allows the program to be developed further.

TABLE 1. Database Test case for staff dashboard and login

Test	Test Data - Login	Expected Result
1.	Login with admin user name and password	open admin dashboard
2.	Login with admin user name and password	open staff board
Test	Test Data – Staff Dashboard	Expected Result
3.	Click on customer name in schedule list in staff dashboard	display customer name with details on right box
4.	Click on service name in schedule list in staff dashboar	display service name with schedule details on right box
5.	Select logout from drop down menu on top right side	logout from account and redirect to login page

TABLE 2. Database Test case for booking

Test	Test Data – Customer Booking	Expected Result
1.	Click book now button on main page	display selected service with details on right side windows
2.	Click submit button without entering any data in input box	display message – fill the required field
Test	Test Data – Admin Dashboard	Expected Result
3.	Click pencil icon button from service list	Display service details in right side windows
4.	Update service and press submit	service update and display message and empty the input box
5.	Click pencil icon button from booking list to update	Modify or edit data and press submit button. Display message “updated successfully”
6	From Invoice click search/display icon button	display invoice with all details
7	Add Schedule – leave one input box empty	display info message “The value required in red color”

9 FURTHER DEVELOPMENT POSSIBILITIES

There are many possibilities to implement a new functionality in future. All the functionality has not been completed yet. As this is a beta version of the booking system, a stable version of the booking system can be done in future after a performance review. The Booking system and admin module are still missing certain user interface such as a displaying service and an available space in calendar view, adding a value added tax and a discount in invoices, displaying places where a service or event is happening, giving a membership or coupons for discount, services images/avatar. Nowadays many websites use Google maps to display their location or event location and such functionality can be implemented in the future for this booking system.

In future the booking system also develop as a stable stand-alone version and plugin for famous content management system (CMS) such as WordPress. Because WordPress is widely used content management with many features like social media management, content management and search engine friendly URL.

10 CONCLUSION

The aim of this thesis was to develop a booking plugin for a front-end and admin module for a stand-alone booking system. The specified requirements were achieved as planned at the beginning.

This thesis work was interesting and challenging because most of the things were new to me. The Project started from very beginning such as database design and making a paper prototype. At the beginning I was not sure that how long it will take and what will be the outcome. But most of technologies, I studied which were used to develop this project. Some small problems arise but after asking for help on a discussion forum such as stackoverflow.com, the project work went smoothly.

The main problem of the project was to upload on the main server, as the project was developed in an old version of the local server. But finally, it uploaded on a web server and is working as planned. At the beginning of project development, I thought using a modernizer might be a problem because I had never used a modernizer in my previous projects during my studies. During the project, I learned a new framework modernizer, PHPUnit and used it successfully in the development.

The Objective of the booking system development was to create standard code which can be reused later. In all, the project was successful and booking system's beta version was ready. The main objectives were reached and the test user is satisfied with the outcome and found the booking system very easy to understand. The development will continue immediately after the WordPress plugin and widget specification have been finalized.

REFERENCES

1. Rahman, S.F. 2014. Jump Start Bootstrap. VIC Australia: SitePoint Pty Ltd.
2. Ruvalcaba, Z. 2012. murach's HTML5 and CSS3 JavaScript DOM jquery PHP MySQL with Index. Frenso, CA: Mike Murach and Associates, Inc.
3. Bootstrap Download. 2014. Date of retrieval 12.09.2015.
<http://getbootstrap.com/getting-started/#download>
4. PHPUnit. 2015. Date of retrieval 10.10.2015. <https://phpunit.de/>
5. JQuery – What is jQuery. 2014. Date of retrieval 12.09.2015
<https://jquery.com/>
6. PHP 5 Multidimensional Arrays. 2013. Date of retrieval 04.04.2014
http://www.w3schools.com/php/php_arrays_multi.asp
7. mysqli:select_db. 2013. Date of retrieval 18.09.2015
<http://www.php.net/manual/en/mysqli.select-db.php>
8. PHP 5 MySQLi Functions. 2014. Date of retrieval 19.09.2015
http://www.w3schools.com/php/func_mysqli_query.asp
9. Modernizer.2015. Date of retrieval 12.09.2015 <https://modernizr.com/>
10. Server-side processing.2014. Date of retrieval 12.09.2015.
<http://datatables.net/manual/server-side>
11. Webpage performance.2014. Date of retrieval 11.11.2015.
<http://www.webpagetest.org/>