

GDS Software Engineer Project Technical Document

Author: Tan Teck Wee Marchand

Contact Number: 93899790

Email: marchandtan89@gmail.com

Introduction

This is a simple technical write up on the Room Booking Portal Project. The project was developed with XAMPP server framework, choice made due to having previous experience on using it so as to be able to focus on developing the use cases to fulfil project completion within the time allotted.

The portal is also stylised with simple default bootstrap styles, which has a responsive layout design to cater for mobile screens display.

As deploying on Docker seems to be non-trivial to setup for the case of xampp and due to time constraints, I have added a download link for the xampp client I used in the github repository instead and a section on setting up the project on a windows platform in this report. However should that be too cumbersome, I am okay to bring my laptop down for project evaluation if that's preferable!

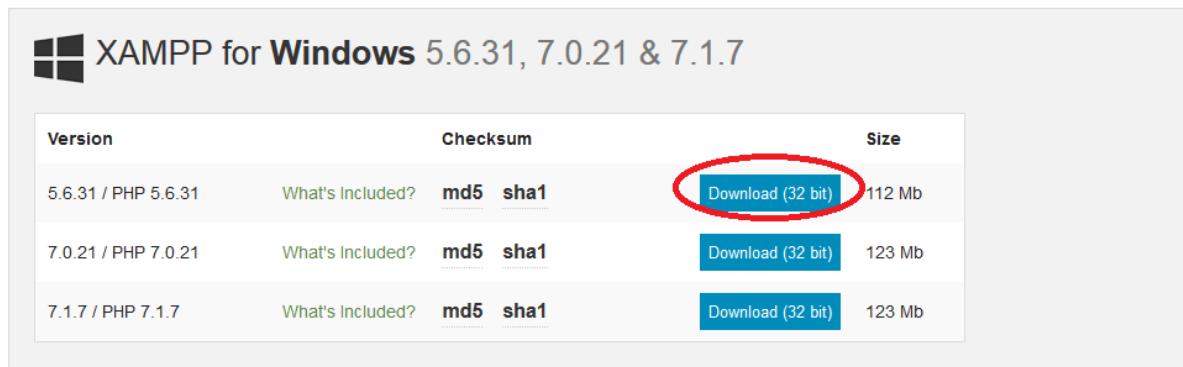
Framework/Languages used

- XAMPP server framework for windows. (A package of Apache HTTP server, Mysql, Filezilla, Mercury, Tomcat)
- Javascript and JQuery
- Html
- Php
- Bootstrap js and css

Installation and setup

Download apache xampp from the link <https://www.apachefriends.org/download.html> as in the download link text file in the repository. I personally chose the 5.6.31/php 5.6.31 version.

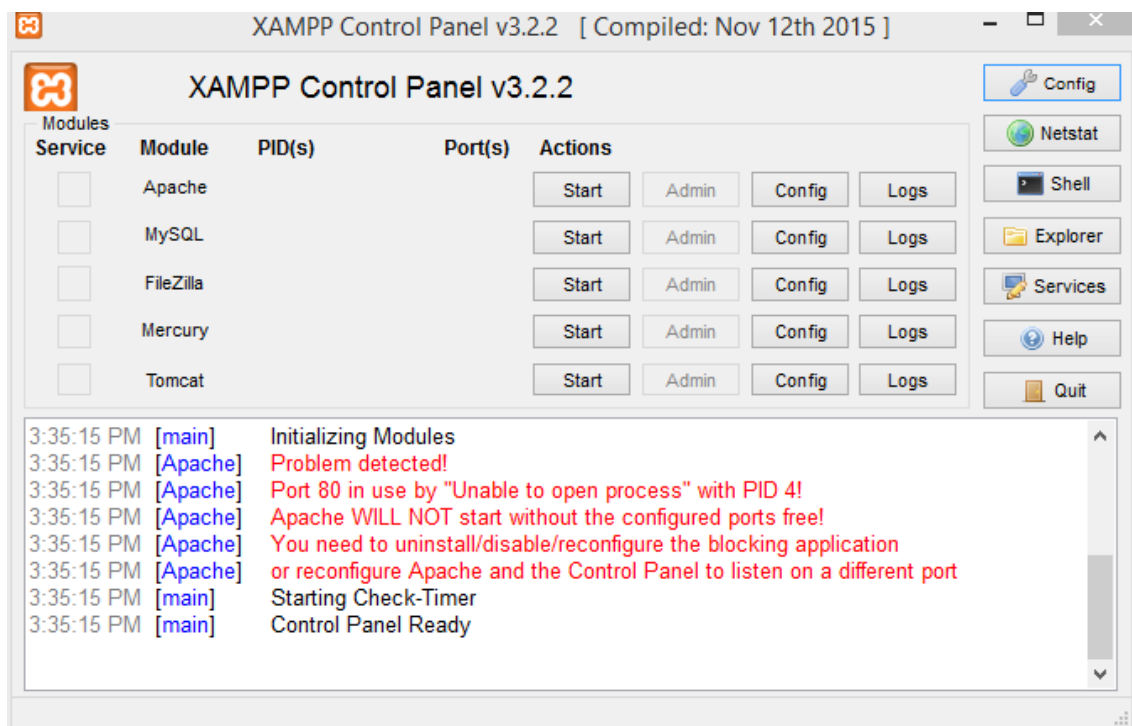
XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.



XAMPP for Windows 5.6.31, 7.0.21 & 7.1.7

Version	Checksum	Size
5.6.31 / PHP 5.6.31	What's Included? md5 sha1	Download (32 bit) 112 Mb
7.0.21 / PHP 7.0.21	What's Included? md5 sha1	Download (32 bit) 123 Mb
7.1.7 / PHP 7.1.7	What's Included? md5 sha1	Download (32 bit) 123 Mb

Once downloaded just install. When installation finished, run the XAMPP Control Panel.



XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

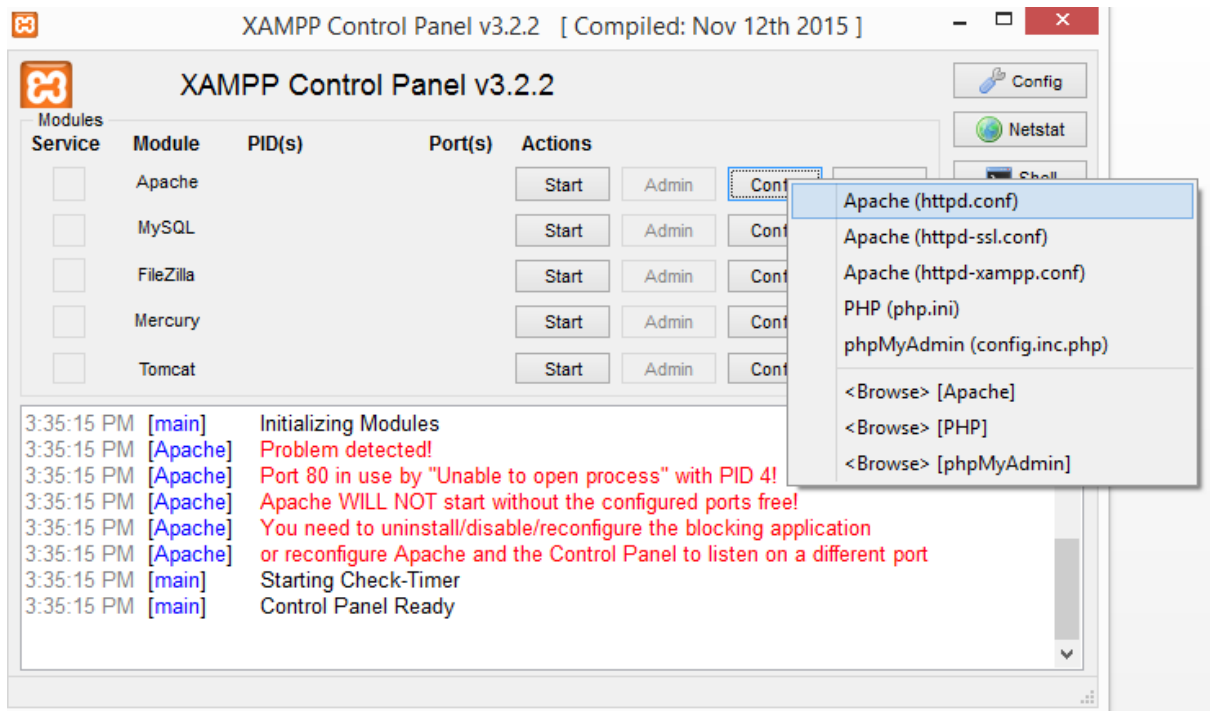
XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

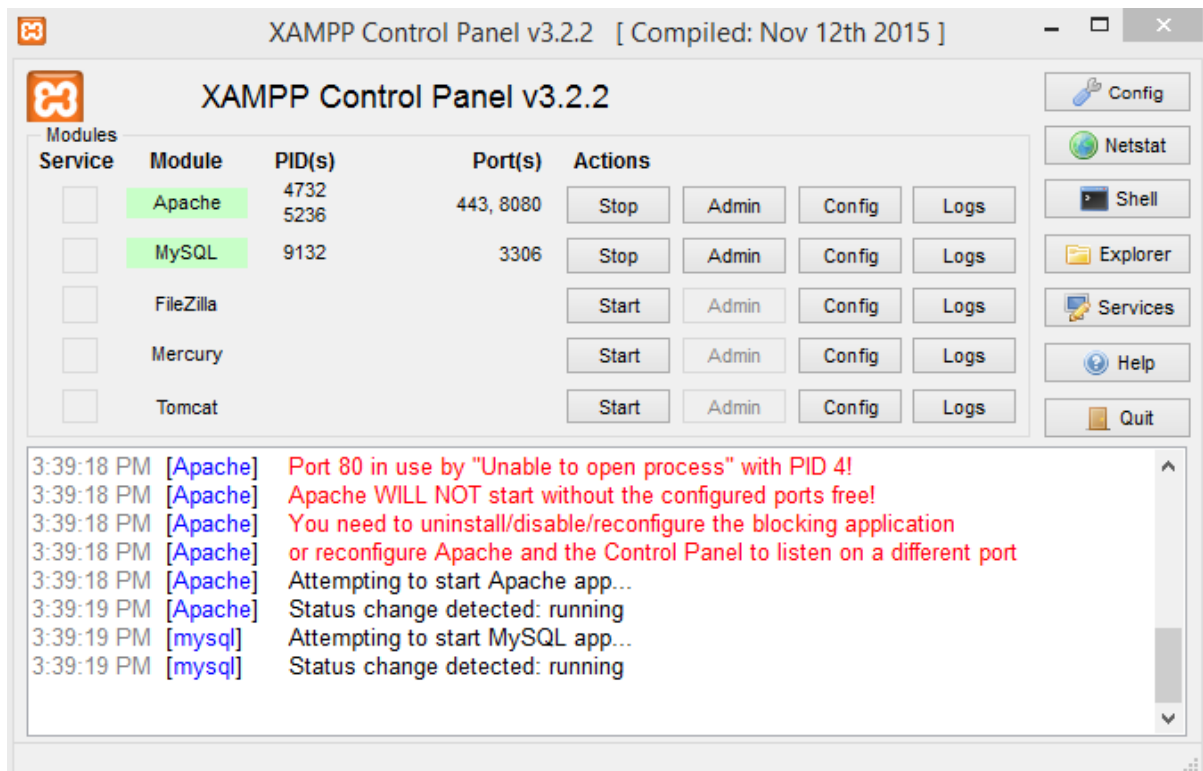
Log window:

```
3:35:15 PM [main] Initializing Modules
3:35:15 PM [Apache] Problem detected!
3:35:15 PM [Apache] Port 80 in use by "Unable to open process" with PID 4!
3:35:15 PM [Apache] Apache WILL NOT start without the configured ports free!
3:35:15 PM [Apache] You need to uninstall/disable/reconfigure the blocking application
3:35:15 PM [Apache] or reconfigure Apache and the Control Panel to listen on a different port
3:35:15 PM [main] Starting Check-Timer
3:35:15 PM [main] Control Panel Ready
```

If you see the above error saying that Port 80 is in use, click Config under Apache.



Select Apache(httpd.conf). It will open up the config text file. Find the line that says Listen 80 and change 80 to a port of choice: eg 8080. Save and close. Now just click Start for Apache and MySQL and you will see the below:

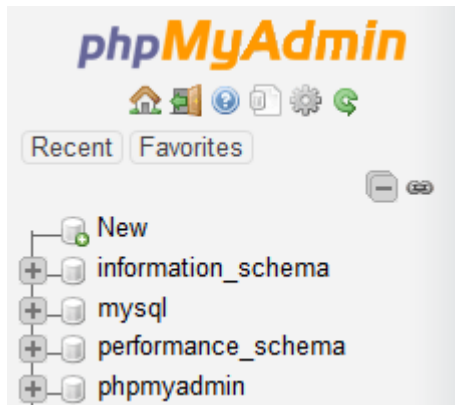


Now our webserver and database is up, and can be explored in a web browser with addresses:

<http://localhost:yourport/> for the webserver main address


<http://localhost:yourport/phpmyadmin/> for managing MySQL database.

Let's go to <http://localhost:yourport/phpmyadmin/> and set up the project database.

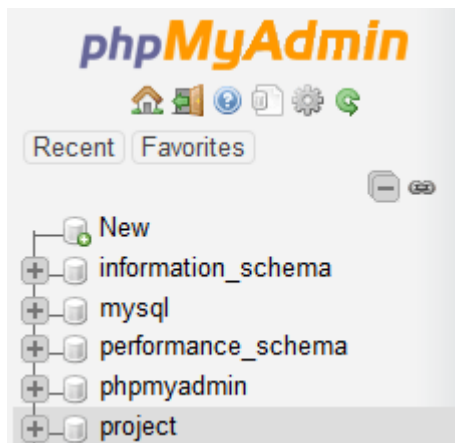


In the left panel of the page click on New.

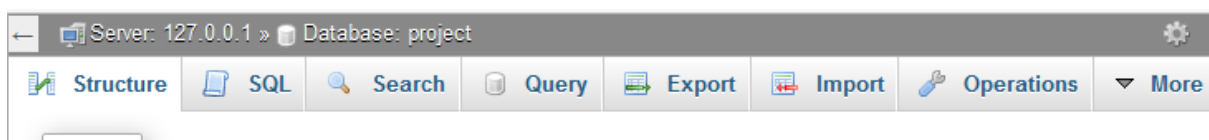
Databases

 **Create database** 

To the right of the page, enter project as name of database and just click Create.



Now go back to the left pane and click on project database which was just created.



To the right of the page, on the top menu bar, click Import.

Importing into the database "project"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: No file selected. (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file:

Click on the Browse button and browse to the `project.sql` file as given in the project repo.

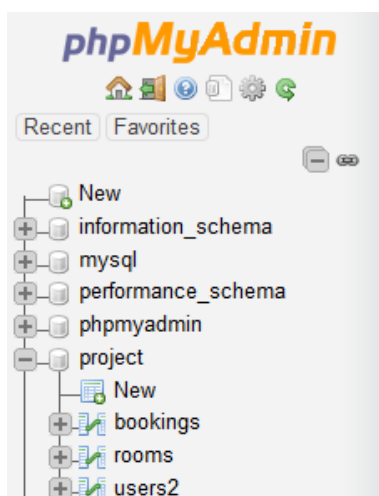
Format:

Format-specific options:

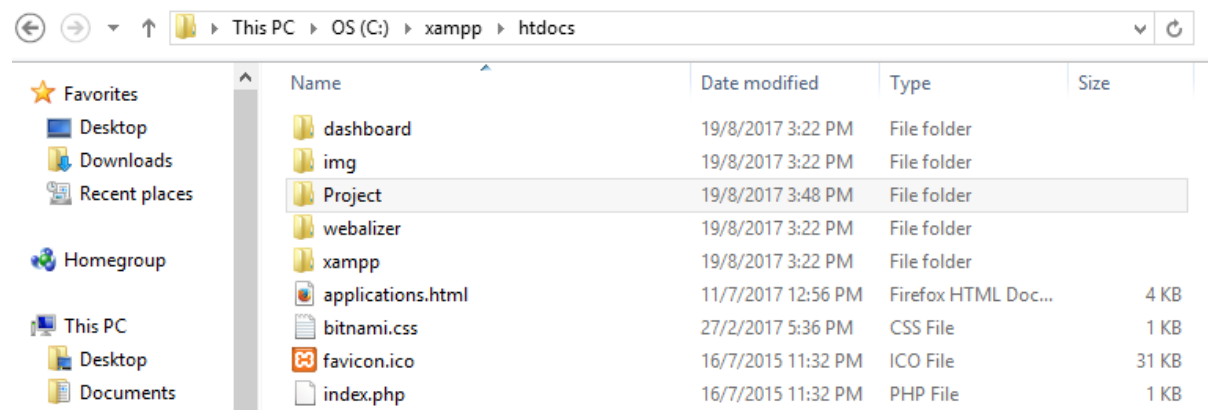
SQL compatibility mode:

☒ Do not use `AUTO_INCREMENT` for zero values

Once done scroll all the way down and when u see the above Go button, click on it. Now you can expand the project database in the left panel and see the below tables have been imported.



Now our database is set up, proceed to copy the Project folder from the repo into the location of your XAMPP htdocs folder (usually in C:\xampp\htdocs)



Now we are ready to go! Just go to the address

<http://localhost:yourport/Project/index.html> to start viewing the project!

Database (Mysql)

This project uses mainly only 3 tables under the database name [project](#). Tables are described below:

users2

The table to describe users of the portal. It's called [users2](#) because my computer crashed once while working on the project and the original table it was based on got corrupted and the table name [users](#) is unusable. The project has been edited to reference this new table instead. There are 3 Columns in this table:

- username – User names
- password – User passwords
- isAdmin – A flag indicating if the user is an administrator, set to false by default when registering new users, only the database admin can create accounts with administrator rights.

rooms

The table used to describe rooms. Rooms can only be registered by administrator user accounts, which will be described in later sections. There are 2 Columns in this table:

- number – Room number
- capacity – Room capacity

bookings

This table describes bookings. There are 5 Columns in this table:

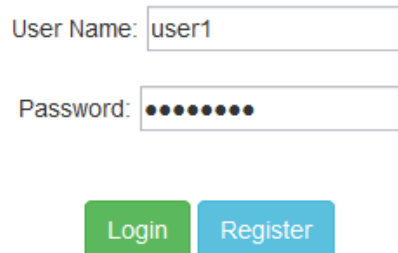
- id – Booking ID, auto incremented
- username – username of the user who made the booking
- roomnumber – room number that was booked
- start – start date time of the booking
- end – end date time of the booking

Site Pages

In this section the portal site's technical aspects will be briefly described based on each page that the user will see.

Main Page: Index.html

Room Booking Portal



User Name:

Password:

index.html

- Main login page. Also handles user registration with a click of a button.

action.js

- Basic empty field detection
- cookie username logging.

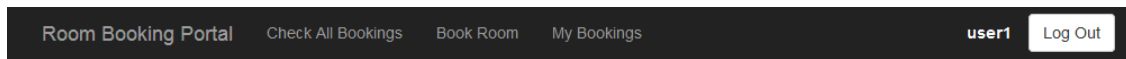
login.php

- Handles user logins
- Input:
 - username
 - password
- Queries:
 - `SELECT * FROM users2 WHERE username='$username' AND password='$password'`
- Output:
 - ok
 - error
 -

registerUser.php

- Registers new users but checks for existing usernames to avoid repeat.
- Input:
 - Username
 - password
- Queries:
 - `SELECT * FROM users2 WHERE username='$username'`
 - `INSERT INTO users2 values ('$username','$password','FALSE');`
- Output:
 - Username already in use!
 - ok
 - error

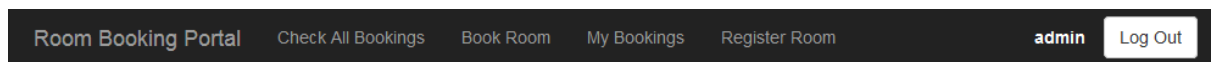
User Page: userPage.html



Welcome! Please select an option in the nav bar above!

userPage.html

- Main portal page for navigation. If logged in with admin account (user:admin,password:admin) in index.html, navigational bar will display one more option for Register Room. (Navigational bar features are present/repeated in every page onwards, but will only be described here for brevity.)



userPage.js

- Set user name in navigation bar with cookie username information
- Checks user name is administrator with checkIsAdmin.php, and sets visibility of Register Room option accordingly

checkIsAdmin.php

- Checks user row from database, see if isAdmin field is 1 to determine administrator account.
- Input:
 - Username
- Queries:
 - `SELECT * FROM users2 where username='$username';`
- Output:
 - ok
 - not admin

Check All Bookings Page: checkBookPage.html

Room Booking Portal

Check All Bookings

Book Room

My Bookings

user1

Log Out

Start Date:

07/26/2017

Start Time:

12:00am

End Date:

07/28/2017

End Time:

11:00pm

Check Booking

Booking ID	User	Room Number	Start	End
5	user1	2	2017-07-28 02:30:00	2017-07-28 16:30:00
6	user1	0	2017-07-28 02:30:00	2017-07-28 16:30:00

checkBookingPage.html

- Page to check bookings made previously against specified date time.

checkBooking.js

- Simple empty fields check.
- Links and usage of external jquery libray jonthornton-jquery-timepicker to integrate intuitive date and time picker

Start Date:

Book

August 2017

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Start Time:

12:00am

12:30am

1:00am

1:30am

2:00am

2:30am

getBookings.php

- Fetches data from bookings table
- Input:

- Start – start datetime
- End – end datetime
- Queries:
 - `SELECT * FROM bookings where start BETWEEN STR_TO_DATE('$start','%d-%m-%Y %T') AND STR_TO_DATE('$end','%d-%m-%Y %T') OR end BETWEEN STR_TO_DATE('$start','%d-%m-%Y %T') AND STR_TO_DATE('$end','%d-%m-%Y %T') OR start >= STR_TO_DATE('$start','%d-%m-%Y %T') AND end <= STR_TO_DATE('$end','%d-%m-%Y %T');`
- Output:
 - No results
 - Query results
 - Error

Book Room Page: bookingPage.html

Room Booking Portal	Check All Bookings	Book Room	My Bookings	user1	Log Out
---------------------	--------------------	-----------	-------------	-------	---------

Room: Room No: 1 Capacity: 15 ▾

Start Date: 08/03/2017

Start Time: 12:30am

End Date: 08/03/2017

End Time: 2:00am

Book Room

bookingPage.html

- Page for booking rooms.

booking.js

- Simple empty field detection
- Uses the same datepicker and timepicker jquery utilities as checkBookingPage

bookRoom.php

- Checks if a previous booking entry is overlapped in this booking entry, and inserts booking entry should there be no overlaps.
- Input:
 - Username
 - roomnumber
 - start – start datetime
 - end – end datetime
- Queries:
 - `SELECT * FROM bookings where roomnumber='$roomnumber' AND(STR_TO_DATE('$start','%d-%m-%Y %T') BETWEEN start AND end OR STR_TO_DATE('$end','%d-%m-%Y %T') BETWEEN start AND end OR STR_TO_DATE('$start','%d-%m-%Y %T') >= start AND STR_TO_DATE('$end','%d-%m-%Y %T') <= end);"`
 - `INSERT INTO bookings (username,roomnumber,start,end) values ('$username','$roomnumber',STR_TO_DATE('$start','%d-%m-%Y %T'),STR_TO_DATE('$end','%d-%m-%Y %T'));`

- Output:
 - Booking date time range overlaps previous bookings. Please choose another date time range
 - ok
 - error

My Bookings Page: bookingRecordPage.html

Room Booking Portal	Check All Bookings	Book Room	My Bookings	user1	Log Out
---------------------	--------------------	-----------	-------------	-------	---------

Booking ID	Room Number	Start	End	Manage
5	2	2017-07-28 02:30:00	2017-07-28 16:30:00	Cancel Booking
6	0	2017-07-28 02:30:00	2017-07-28 16:30:00	Cancel Booking
8	1	2017-08-01 20:00:00	2017-08-02 19:00:00	Cancel Booking
9	1	2017-08-03 00:30:00	2017-08-03 02:00:00	Cancel Booking

bookingRecordPage.html

- Page to display the user's past bookings. User can click on Cancel Booking to remove entry from database.

bookingRecord.js

- Bootstrap table view with cancellation buttons based on id. Nothing special.

getUserBookings.php

- Fetches user booking records based on username.
- Input:
 - Username
- Queries:
 - `SELECT * FROM bookings where username='$username';`
- Output:
 - Query Results
 - error

cancelBooking.php

- Deletes a booking record from database based on id.
- Input:
 - Id – booking id
- Queries:
 - `DELETE FROM bookings where id='$id';`
- Output:
 - ok
 - error

Register Room Page: roomRegister.html

Room Booking Portal	Check All Bookings	Book Room	My Bookings	Register Room	admin	Log Out
---------------------	--------------------	-----------	-------------	---------------	-------	---------

Room Number:

Capacity:

roomRegister.html

- Administrator access only page for registering rooms. To see this page on navigation bar, login at index.html with (username:admin, password:admin) or any other manually inserted database admin accounts.

registerRoom.js

- Simple empty field check
- Simple jquery isNumeric check

registerRoom.php

- Checks if room number has been registered previously, and inserts new room entry if room number is new.
- Input:
 - roomnumber
 - capacity
- Queries:
 - `SELECT * FROM rooms WHERE number='$roomnumber';`
 - `INSERT INTO rooms values ('$roomnumber','$capacity');`
- Output:
 - ok
 - error