Dimitrios Karanasios

n9389491  Due: 28 November 2016

ifb299 person portfolio

Contents

[One – Registering Online 2](#_Toc465458000)

[Two – Course Booking 2](#_Toc465458001)

[Three – Cancel Course Bookings 2](#_Toc465458002)

[Four – Volunteering/Server Booking Restrictions 3](#_Toc465458003)

[Five – Admin Panel 3](#_Toc465458004)

[One – Contact Form 4](#_Toc465458005)

[Two – Messages (Admin) 4](#_Toc465458006)

[Three – Server History (Admin) 5](#_Toc465458007)

[Four– Bug Fixing 5](#_Toc465458008)

[Five - Cleaning up the Code 5](#_Toc465458009)

Release 1

# One – Registering Online

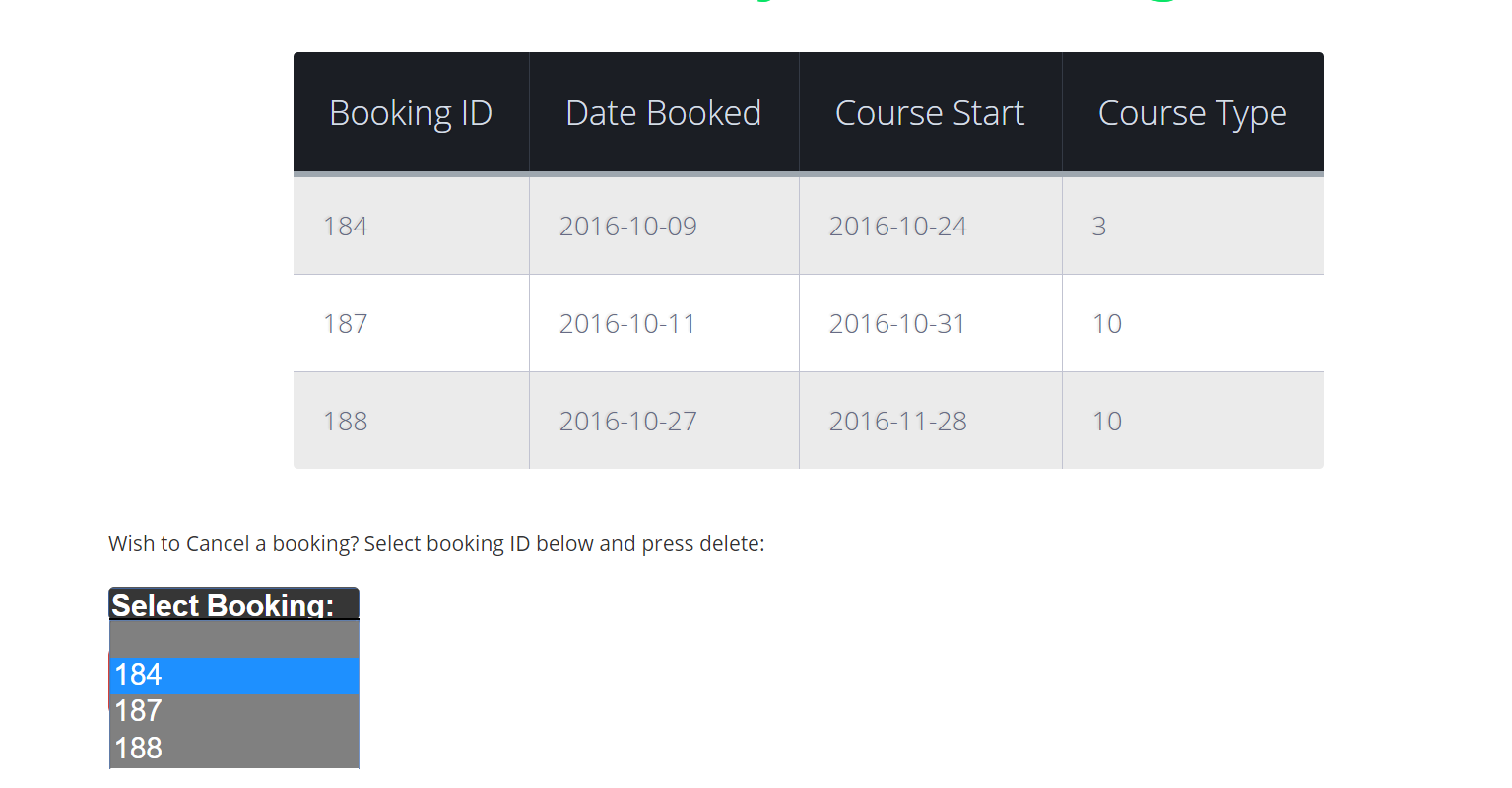


When creating the database, deep consideration was taken in order to know what details were most important and necessary to be filled in by the user for admins to keep on record. The database was created using phpMyAdmin, a free software allowing us to store data easily as well as ensuring the data is being stored securely (including being able encrypt the data).

# Two – Course Booking

The Course Booking is one of the main features of the website. With Mitch and Jon having designed the booking form to be as simple as possible for the user to easily and quickly make a booking, I had worked on the programming with php to take in the input from the form and enter the data into the database. I also added in booking restrictions given by the client including adding a maximum number of people per course, only allowing classes to be booked on Mondays and also restricted a user from double booking on the same date.

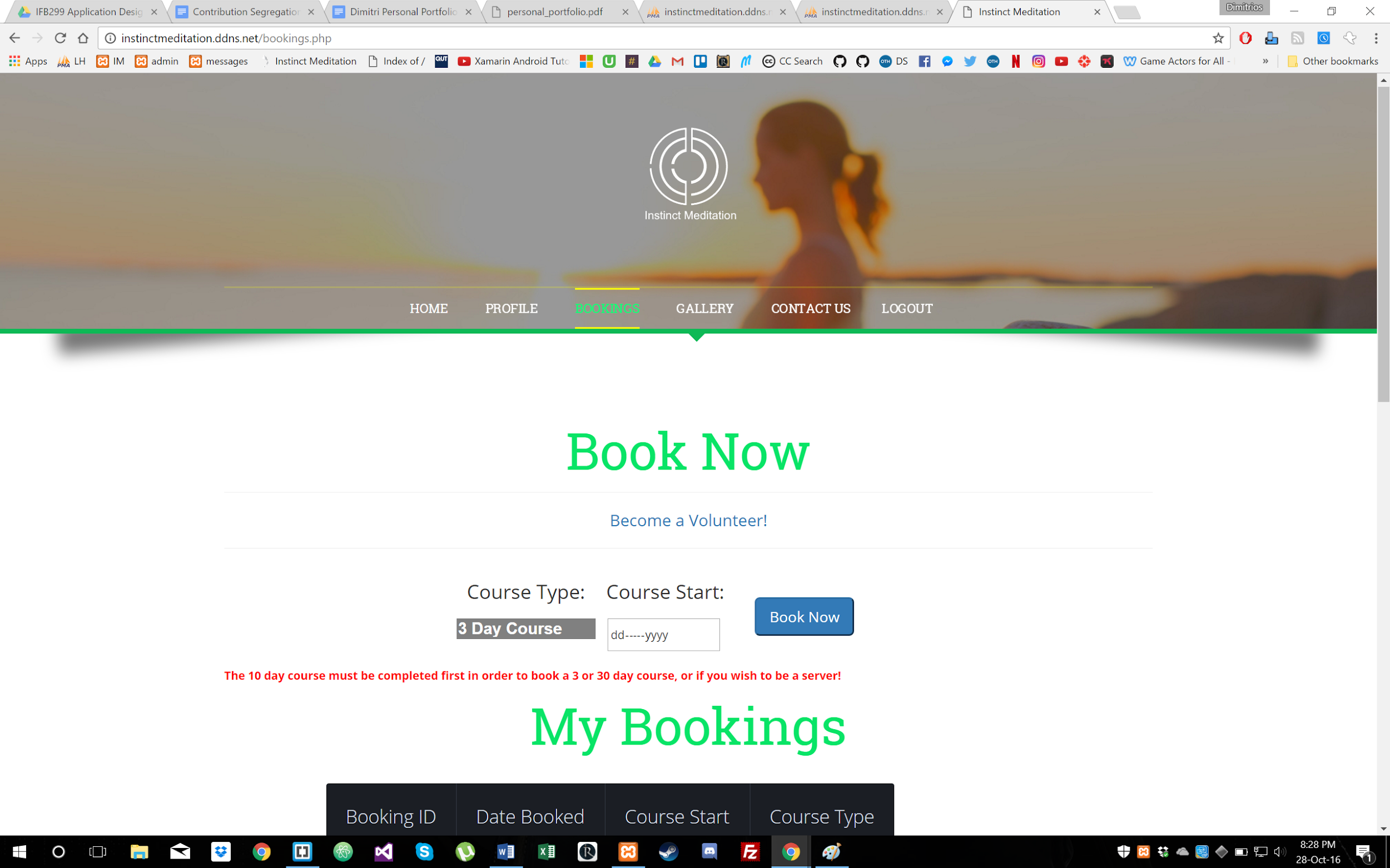
# Three – Cancel Course Bookings

Underneath the Booking History table is a form I added to allow the user to delete any of their bookings. When developing the delete booking feature, I made sure that the drop down menu only consists of options for bookings made by that user and to not allow them to cancel any other user’s bookings. To avoid accidental course booking cancellations by the user, I did some research and found the best option was to add a pop up form for the user to confirm their cancellation. I implemented the pop-up using JavaScript to ask a quick yes or no confirmation question.



# Four – Volunteering/Server Booking Restrictions

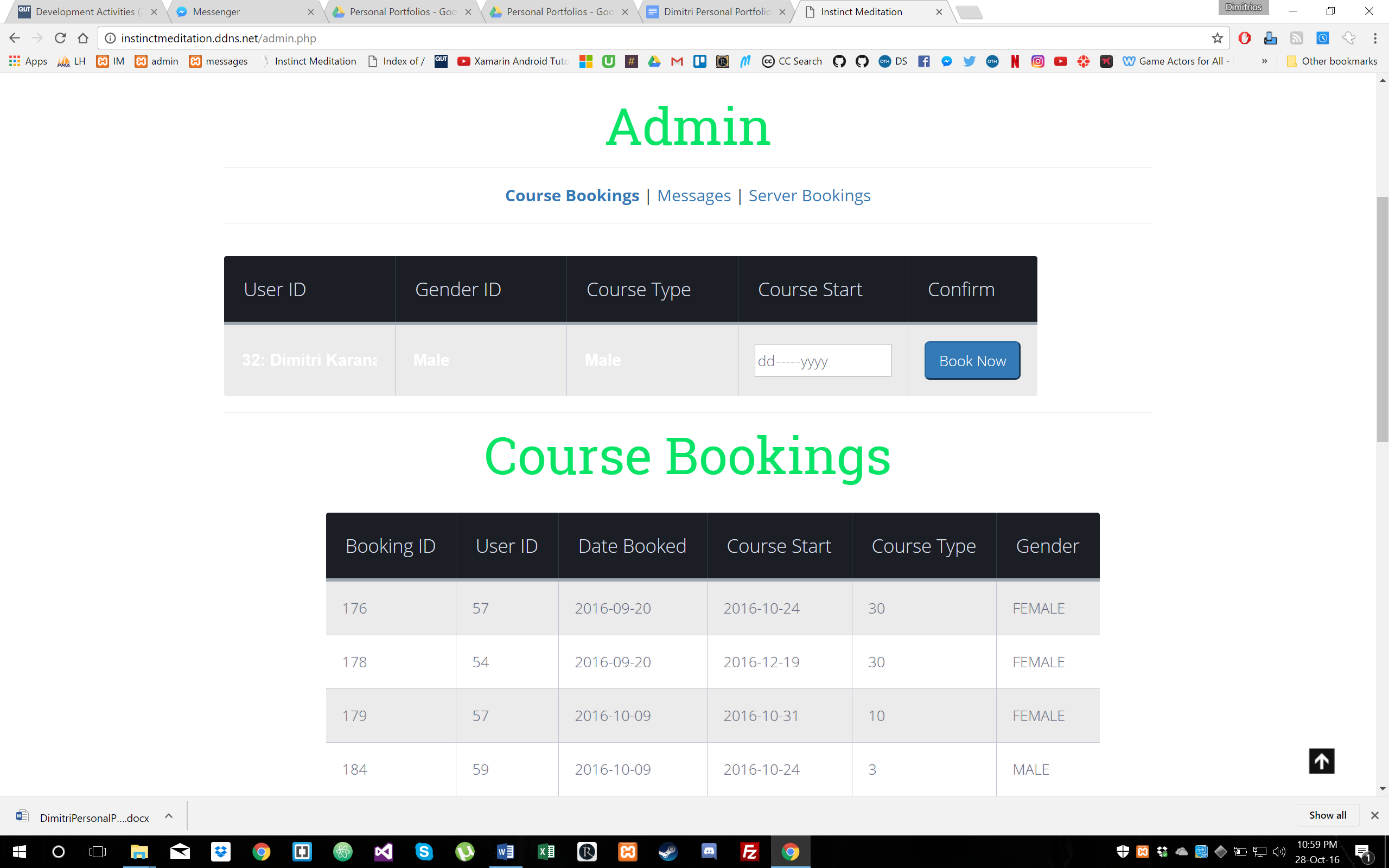
10 day course reqruiements



# 

# Five – Admin Panel

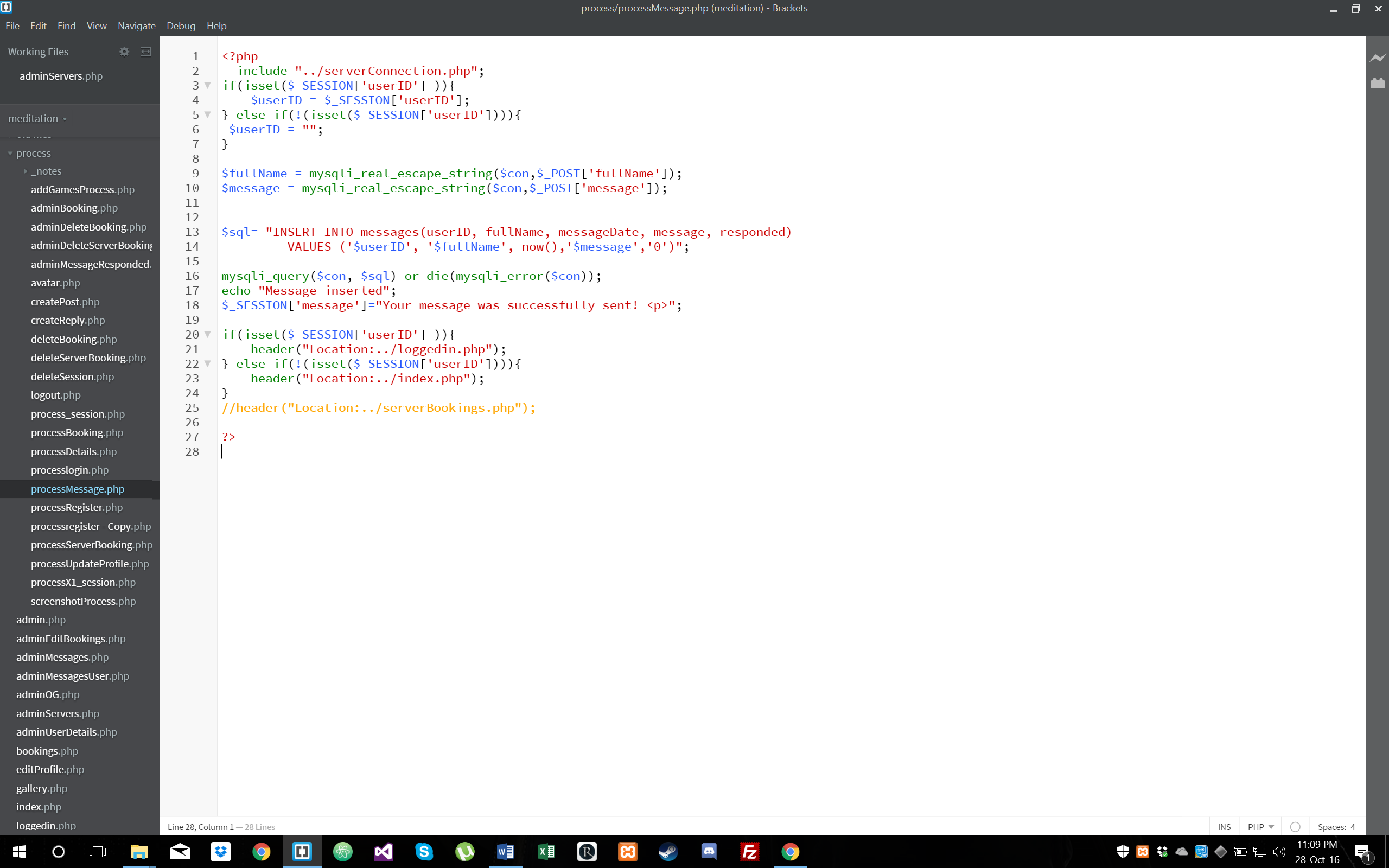
One of the admin privileges is being able to view the full sql database and edit parts of it without actually needing access to the database which can be confusing for some admins to understand how to use. This would also prevent the admin staff from mistakenly deleting or adding data and potentially damaging the database. The admin panels outputs data from the database including user course booking information, messages sent in via the contact form and also the server booking information. I also created a form for the admin to delete any user course bookings/server bookings and also a form to create a user course booking, in the case of requiring to manually make a booking for a user if they were to have any issues on their account and are unable to make a booking themselves.



Release 2

# One – Contact Form

During sprint 3 presentation we showed off the contact form where the user was required to login to be able to have access to the contact form. The client discussed their opinions on wanting users even without an account to have access to the form which was then changed. I worked on making sure that correct data from the form was entered into the database including with the form being auto filled when logged in or when a person manually enters in their information and message.



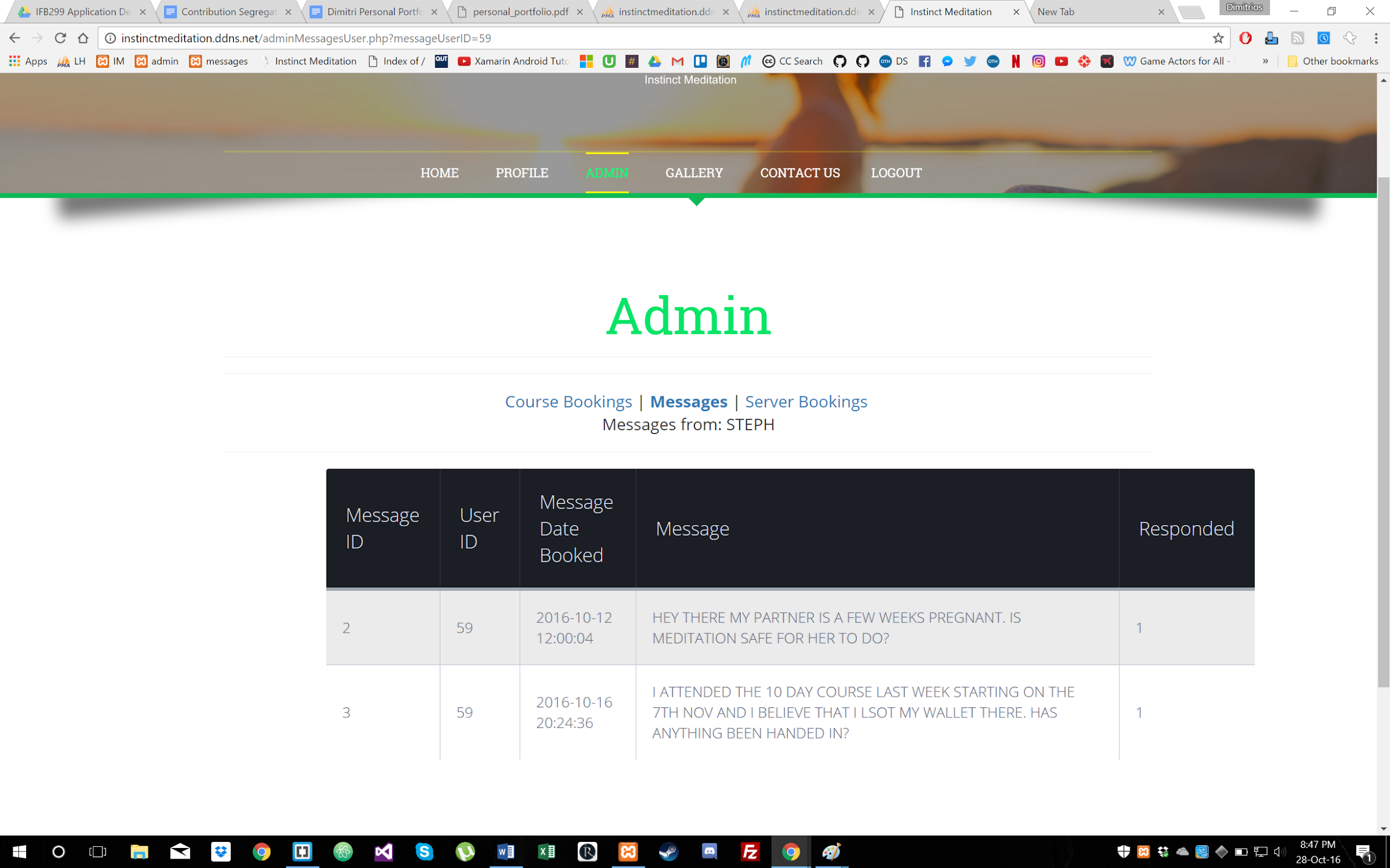
# Two – Messages (Admin)

In the admin panel, the admins are able to see all the messages sent by people (via the contact form). The table shows the message and also has the user-ID of the sender with the ability to click on the user-ID to see the all other messages sent by that user. In order for the admin to be able to understand which messages have been replied to, I implemented a button of which the user can click to toggle whether or not the message has been resolved.

# 

# 

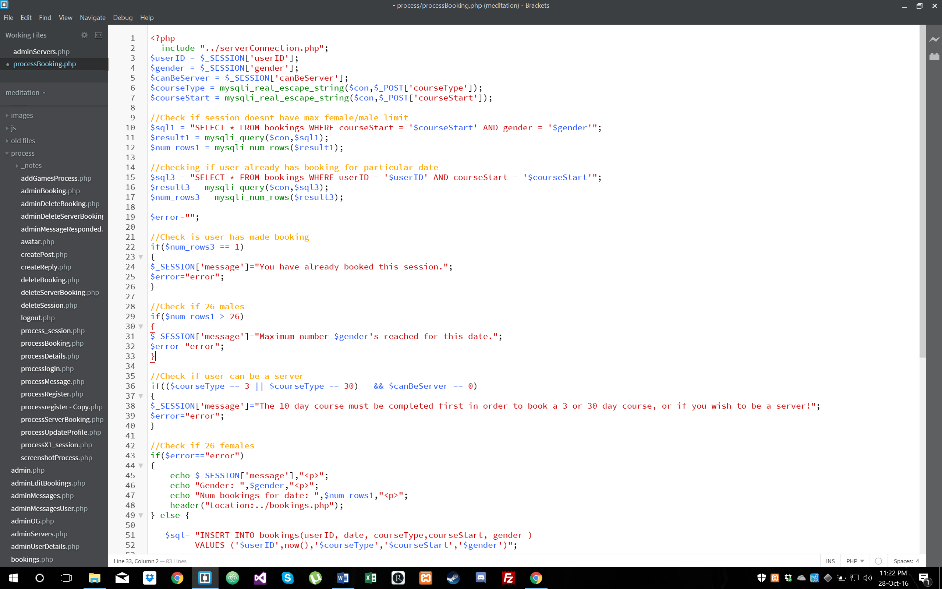
# Three – Server History (Admin)

Another admin privilege I added is the ability to view all Server Bookings made. The admin can see all necessary information for a servers booking from the table in the admin panel. If any more information about the server is required, I developed the ability to click on the server’s user ID to be redirected to a page that will show all information about that person. The admin also has the ability to delete any of the server bookings if required.



# Four– Bug Fixing

With Cameron, Chee and Jon working on testing the website, they were able to identify and list several bugs of which required fixing. Looking through the list of bugs and what happened to cause them, I compared those to my code to see why a certain error is being triggered when it shouldn’t be. The bugs that were fixed include receiving the wrong error message upon making a booking (e.g. being told you already have a booking when the class is full). Another issue that was found was the admins not being able to ‘Un-resolve’ a message from a user’s message in the admin panel.



# 

# Five - Cleaning up the Code

After the development of the site was completed, Cameron and I went through the code and began to clean it up. This involved removing any unnecessary code, adding comments and fixing the formatting/styling of the code (spacing and gaps between functions etc.). Doing this would allow any new developers to be able to easily understand the code when upgrading or making changes to the website.

