

## Web Application Assignment

Due: Wednesday 11:59pm 5<sup>th</sup> of February 2020 – **Assessment Weight: 30%**

---

### Note:

- Include comments for your student ID, Name, and Practical Class Time at the top of each source file created.
  - All instructions given in this document must be followed in order to be eligible for full marks for the Web Application Assignment
  - This assignment is not a group assignment; collusion, plagiarism, cheating of any kind is not acceptable. Submitting your work to your TWA site signifies that the work uploaded is yours. If you cannot honestly certify that the work is your own then do not upload it to your TWA site. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information)
  - All files must be uploaded to your TWA web site before due date.
  - After files are uploaded to your TWA site, all work must also be submitted to vUWS. **Failure to upload to your TWA site and vUWS may result in a non-submit or a significant loss of marks.**
  - **Difference in your TWA site files and vUWS submission will result in a 10% penalty.**
  - Ensure all HTML written is valid. Use <http://validator.w3.org> to confirm before submission
- 

### Assignment Overview:

For this assignment, you are going to be build a basic airline booking system. To do this, you are given a list of files that you will need to create. These will be a mix of HTML, CSS, Javascript and PHP files. Within some of your PHP files, you will be required to connect to and execute SQL statements on a database. This database will be given to you and will be associated only with your TWA site.

Every concept or skill within this assignment has been demonstrated in lecture or has been part of a practical class exercise. You will be combining all these concepts and skills to build your basic eCommerce site.

### Design and Styling

Within lectures and your practical classes you have been given the styling and design that needs to be met either via CSS or via a figure. For this major web assignment though, you must design and style your own site. You can adopt what has been given in your Practical Class exercises or you can start from scratch. Either way, you must design and style your site so it is professional looking. Your chosen design and styling will be assessed.

### Javascript and PHP Validation

All forms or where a user can provide input needs to be validated with both Javascript and PHP. HTML 5 validation will not be accepted as a valid form of validation.

### Web Application Assignment Database

You have been provided with your own copy of the database called cooper\_flights on your TWA site. To access this database, you have to use a username and password.

The following is a generic representation of the connection information to be used when connecting to your eshop database (you do this with PHP code):

Database Name: **cooper\_flights###**  
Username: **twa###**  
Password: **twa###XX**  
Server: **localhost**

Where **###** is your twa site number, and **XX** refers to the first two characters of your TWA site password.

For example, if your TWA site is twa**999**, and your password is **ab**cd7890, then the following would be your connection information:

Database Name: **cooper\_flights**  
Username: **twa999**  
Password: **twa999ab**  
Server: **localhost**

Using this information, you will use similar code to that below to connect to your database:

```
$connection = new mysqli('localhost', 'twa999', 'twa999ab', 'cooperflights999');
if($connection->connect_error) {
    // Echo out if it failed to connect to the database
    echo $connection->connect_error;
}
```

Once connected to your database, you will have access to the **cooper\_flights** database and all its data. Figure 1 presents the Entity Relationship Diagram and the schema of your database.

If wanting, within the assignment files the Schema has also been provided – **cooper\_flights.sql**

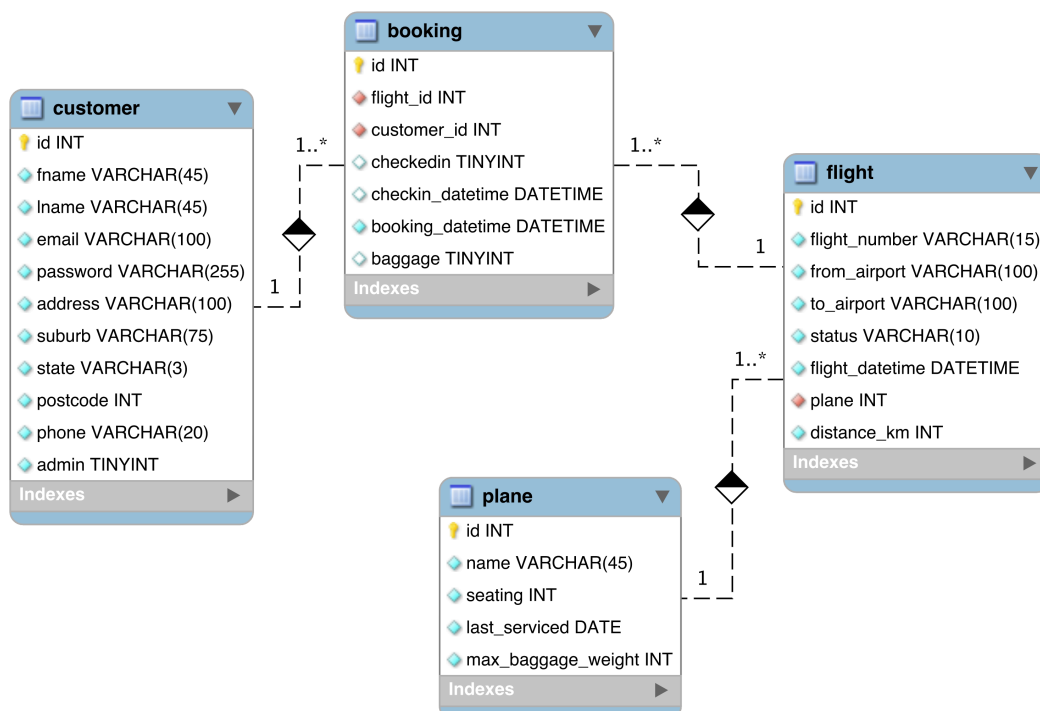


Figure 1

*The cooper\_flights Database Data Dictionary:*


Customer Table			
Column	Description	Type	Required
id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
email	This is the email address of the user. It will be used as their login	VARCHAR(100)	Yes
password	This is an encrypted password. The encryption used is sha256.	VARCHAR(255)	Yes
fname	This is the first name of the user	VARCHAR(45)	Yes
lname	This is the last name of the user	VARCHAR(45)	Yes
address	This is the home street of the user, e.g. 4 Silly Street	VARCHAR(100)	Yes
suburb	This is the home suburb of the user, e.g. Parramatta	VARCHAR(75)	Yes
state	This is the home state of the user, e.g. NSW	VARCHAR(3)	Yes
postcode	This is the home post code of the user, e.g. 2000	INT	Yes
phone	This is a phone number	varchar(20)	Yes
admin	This is a binary value to state if the user is an admin or not. 0 being general user, 1 being admin	TINYINT	No (Default 0)

Booking Table			
Column	Description	Type	Required
id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
flight_id	This is a foreign key to the flight table for the flight booked	INT	Yes
customer_id	This is a foreign key to the customer who made the booking.	INT	Yes
checkedin	This is a binary value to state if the user has checked in or not. 0 if they're not checked in, 1 if they are	TINYINT	No (Default 0)
checkin_datetime	This is the date and time the customer checked in for their booking	DATETIME	No
booking_datetime	This is the date time the customer made the booking	DATETIME	Yes
baggage	The weight of the checked in baggage	TINYINT	No (Default 0)

Flight Table			
Column	Description	Type	Required
id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	Integer	Yes
flight_number	This is the number representing the flight	VARCHAR(15)	Yes
from_airport	This is the airport the flight departs from	VARCHAR(100)	Yes
to_airport	This is the airport the flight arrives at	VARCHAR(100)	Yes
status	This is the status of the flight. There are four stages: staged, open, departed, and cancelled	VARCHAR(10)	Yes
flight_datetime	This is the date and time the flight departs	DATETIME	Yes
plane	This is a foreign key representing the plane being used for the flight	INT	Yes
distance_km	This is the flight distance in kilometres	INT	Yes

Plane Table			
Column	Description	Type	Required
id	This is an auto incrementing number to uniquely identify a table row. You do not insert this number into the database it is determined automatically	INT	Yes
name	The name of the plane	VARCHAR(45)	Yes
seating	The number of seats on the plane	INT	Yes
last_serviced	The date the plane was last serviced	DATE	Yes
max_baggage_weight	This max weight the plane is able to carry for passenger baggage	INT	Yes

### Use of Javascript and CSS Libraries

 **IMPORTANT:** This should only be used by students with prior understanding of these libraries. If used, instruction and support on how to use these libraries will not be provided by your tutor.

Although they have not been presented in lecture or practical classes, if you feel you want to use either the jQuery Library for Javascript or the Bootstrap Library for styling you may. However, you are only allowed to use the code given to you in the **optional-libraries.txt** file. You are not to source bootstrap templates of any kind or jQuery components. If you are found to have done this a misconduct investigation will be raised.

The HTML `<link>` tags within the **optional-libraries.txt** use content delivery networks to provide the sources for jQuery and Bootstrap. You should not have the frameworks uploaded to your TWA site.

## Required Functionality and Files

**Hint:** Write your HTML, CSS first before you do any Javascript or PHP programming. This way you can get your pages designed the way you want and then add the programming logic. This approach will make development considerably easier. Plus, if you fail to implement the Javascript or PHP you can still get marks for the HTML and CSS used to construct the page.

### Navigation

The site must have a main navigation bar. Placement of this navigation bar is up to you. **Every page should have this navigation bar.** Pages which fail to have a navigation bar will lose marks. The navigation bar should display the following links:

- Home – links to **index.php**
- My Profile – links to **profile.php**
  - *This should only be present if the user is logged in*
- New Booking – links to **newbooking.php**
  - *This should only be present if the user is logged in*
- Bookings – links to **bookings.php**
  - *This should only be present if the user is logged in*
- Register – links to **register.php**
  - *If the user is logged in this link should not be present*
- Login – links to **login.php**
  - *If the user is logged in this link should change to 'Logout' (logout.php)*
- Flights – links to **flights.php**
  - *This is only available to logged in users who are admin*

**⚠ Warning:** Be aware of what pages require the user to be logged in. If an anonymous user tries to access these pages they are to be redirected to **login.php**

### Style and Design – styles.css

The styles.css file is your master CSS stylesheet for your site. Although you are allowed embedded and inline styles, a focus on external styles should be maintained. For maximum marks, your style and design should be responsive. This means it can be viewed both on a mobile screen and a desktop. Use the device preview available in Google Chrome under Developer Tools.

### Javascript – actions.js

This file is to hold all your javascript required for your pages; all your form validations or actions javascript will perform.

## Home page – index.php

**Hint:** The listing of flights on this page can be reused across many pages with slight changes. It is advised to solve this first before attempting the other pages so you can reuse your code.

This file is the home page to your site. It is to list all flights for the next month. In this listing each flight should present:

- Flight Number
- Depart date and time
- Status
  - Status has an icon associated with each status.
    - Refer to the image names: staged.svg, open.svg, departed.svg, and cancelled.svg
- Departure Airport
- Destination Airport
- The Plane
- Flight Distance

Be mindful when presenting these listings as your site needs to be mobile responsive. This means a table may not be the best for presenting this information. Figure 2 presents a **sample** listing of flights for inspiration. You are not required to create figure 2, it is only presented for inspiration.

<b>Flight: QF1986</b> Cape York -> Cairns (777km) - Cessna Citation Mustang	<b>Departed</b> ✈ 9:30am 02/01/2020
<b>Flight: QF1769</b> Cape York -> Darwin (1285km) - Cessna CitationJet 525	<b>Open</b> ● 5:30pm 02/01/2020
<b>Flight: QF1986</b> Cape York -> Cairns (777km) - Cessna Citation Mustang	<b>Staged</b> ■■ 9:30am 03/01/2020
<b>Flight: QF1888</b> Cape York -> Brisbane (2159km) - Embraer Phenom 100	<b>Cancelled</b> ❌ 12:30pm 03/01/2020

Figure 2

## My Profile – profile.php

This page presents the currently logged in users' details within a form (except their user id, if they are admin and their password). If the details have changed the user can update details and click submit to perform a **postback** to save the modified details to the database.

This page requires the following validations before the **postback** can occur:

- Basic form validation so all fields are required
- The email address is a valid email address form
- Post code must be 4 digits
- State must be 3 characters or less (Hint: use a drop down)

## New Booking Page – newbooking.php

This page allows the user to select a flight to book. You can present the flights as a list (from home page), with an additional link to book. However, **the link should only be present if there is room to book a seat**. This is because each plane has a seating capacity. To ensure a plane is not over booked you will need to compare the number of bookings for each flight and compare it against the number of seats on a plane. If the seating capacity has been reached you are to present 'Full' and remove the book flight link, otherwise you can present the number of seats left and a link to open a book flight form. This form can be on the same page or a separate page. For instance, **newbooking-form.php**.

The booking form requires the following fields:

- The flight number prepopulated from the previous selection from flights
- A read only amount of a standard fee - \$455.00
- Credit card number
- Credit card name
- Credit card expiry

Although the form has credit card fields they are not stored and are only for presentation. However, before submission to complete the booking, the fields are required.

Once submitted, a booking must be inserted into the database with the booking date and time being the date and time the form was submitted. The flight\_id and the customer\_id must also be inserted based on the flight selected and the currently logged in user.

## Bookings Page – bookings.php

This page must present all the bookings the customer has made. Each booking should have the following:

- Flight Number
- Flight Status
- Destination Airport
- Date booked
- Flight departure date and time
- Checked in status.
  - This should display the number of days before the customer can check in. However, if the flight is within 48 hours a link allowing the customer to check in (**checkin.php**) needs to be presented. Once checked in a checked in notice should be displayed.

## Checkin – checkin.php

⚠ **Warning:** Be aware of what pages require the user to be logged in. If an anonymous user tries to access this page they are to be redirected to **login.php**. Also, they cannot reach this page unless the flight selected is to depart within 48 hours.

Based on the flight selected on the bookings page that is eligible for check in this page is to present a form with:

- Flight Number
  - Presented as text not an input element
- Allowable baggage weight
  - Presented as text not an input element. The allowable weight is determined by the max baggage weight of the plane for the flight divided by the planes seating. For instance, the Embraer Phenom 100 has a seating of 4, and a max baggage of 100kg.  $100/4 = 25\text{kg}$  allowable baggage weight per customer.
- The customers baggage weight
  - This weight is checked against the allowable baggage weight otherwise a notice is presented and the customer cannot checkin.

Once the form is submitted the booking for this flight is updated to checked in, and the date and time the form was submitted is the date and time the customer checked in. Once updated the customer is then redirected to the bookings page again.

## User Registration – register.php

This is the user registration page.

To process this form, you are to use a **postback** to submit the data back to **register.php**. Once the form is valid, a new user is saved to the database. They are then automatically logged in with a session where they are then redirected to the home page (**index.php**).

Be aware, the **id** field is not to be entered in by the user. The database automatically populates this field.

Although it won't be the case for production environments, the admin field is required to be present in your registration form. This will allow you to either create a customer or an admin user that will also be able to administrate flights.

When processing the password field, it must be hashed (encrypted). The encryption algorithm to be used is **sha256**. The following code outlines how to use the **hash** function to perform this encryption algorithm:

```
$password = "234jk1dd";
$hashedPassword = hash('sha256', $password);
// $hashedPassword now contains:
// a5d95f8ebc8d8d592cfe772d33d8833909f526233d9ae12c32ada2bbea6a0bba
```

This form needs to be validated before submitting. The required validations are:

- All fields are required
- First name and last name must start with a capital letter
- The email address must be a valid email address



- Post code must be 4 digits only
- State must be 3 alphabet characters or less
- Password must be 8 characters long and must include at least 1 number [0-9]

### User Login – `login.php`

This page will hold a login form containing a username (the user's email) and a password field. Both 'customer' and 'admin' users can login through this login form. Remember, when checking if a user's password is valid you will have to encrypt it before you can check if the password is correct. This encryption is the same process found in `register.php`.

A **postback** must be used to login the user. If all credentials are valid, a user session is created and then the user is redirected to the home page (`index.php`). If the credential are invalid, the user is presented with an error informing them the username and password entered is incorrect. Remember to store the **admin** field in your session, as this will be used throughout the site and in the navigation bar to determine if an 'admin' is shown the Flights link (`flights.php`).

### Logoff – `logout.php`

This page is not a presentation page. When a user navigates to this page their user session is destroyed. Once destroyed, the user is then redirected to the home page (`index.php`)

### Flights – `flights.php`

**⚠ Warning:** Be aware of what pages require the user to be logged in. This page is only for those users who have admin status. All other users are to be redirected to `login.php`.

This page is to list all flights for the current month. For each flight there should be an option allowing admin to cancel a flight that has not departed or is open for check in yet.

### Submission Instructions

All files are required to be on your TWA site under the `/assignment1` directory and also submitted **as a zip** on vUWS before the submission due date. If there is a difference between the submissions, **you will receive a submission penalty of 10%. Also, if you fail to submit to your TWA site, but submit to vUWS, your assignment cannot be tested on your TWA site. This will result in significant loss of marks – you cannot achieve more than 50% as it cannot be tested locally.**

If you cannot honestly certify that the work is your own then do not submit. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).

##### IMPORTANT! #####

**The submission link on vUWS will not be shown until you have agreed that you have uploaded all files on your TWA site and they are the same as your submission to vUWS.**

#####

### Submission Checklist

As a guide, here is the list of files needed to be on your TWA site and vUWS before the submission due date are:

- ☐ `styles.css`
- ☐ `actions.js`
- ☐ `index.php`
- ☐ `profile.php`
- ☐ `newbooking.php`
- ☐ `bookings.php`
- ☐ `checkin.php`
- ☐ `register.php`
- ☐ `login.php`
- ☐ `logoff.php`
- ☐ `flights.php`
- ☐ `staged.svg`
- ☐ `open.svg`
- ☐ `departed.svg`
- ☐ `cancelled.svg`