

Project Report

ONLINE CONCERT TICKET RESERVATION

CS6314 Section 0U2
Web Programming Languages
Instructor: Dr. Nurcan Yuruk

AUGUST 9, 2017

By:

Juanjuan Zhao
Anqi Chen
Qianying Lin

Table of Contents

Introduction	4
Project Description	4
Guest	4
Catalogue	4
Search for concert	4
Filter concert by time range	5
User	5
Register/Login	5
Select seat	5
Add to Cart	6
Check out	6
Order history	7
Logout	7
Administrator	7
Add new concert	7
Update concert	8
Delete concert.....	8
Project Assumption	8
Database Design.....	9
Language/Framework.....	12
Screenshots	13
Work Division	15

List of Figures

Figure 1 Home page	13
Figure 2 Concert information page	14
Figure 3 Register page	14
Figure 4 Login page.....	15
Figure 5 Select seat page.....	16
Figure 6 Shopping cart.....	17
Figure 7 Order history	17
Figure 8 Add new concert.....	18
Figure 9 Edit concert page.....	19
Figure 10 Upload image page	19

Introduction

In this project, we designed a web application that allows users to order concert ticket online. The aim of the project is to provide a web interface for people who want to get the information about concerts that are currently displaying as well as information about upcoming concerts. The website makes it easy for people to view various concerts in the catalogue, choose seat and order the tickets for their choices.

Project Description

Guest

Catalogue

When not logged in, everyone who browses our home page will be able to view a list of all concerts that are currently displaying as well as upcoming concerts. On the homepage, we showed the concert name and picture. By clicking “Learn more” for each concert, they will be able to know the details of each concert, including price, player, description, and show time. There are four concerts shown on each page, people can go to different pages by clicking the page number button at the bottom of the home page. They can also go to previous or next page by clicking the “<<” and “>>” button.

Search for concert

By entering the keywords in the input field on the top of the homepage and clicking the button on the right, a list of concerts that have the keywords appear in their name, player or description will be shown on the page. Others that do not include the keywords will not be shown. There could be one or more results that include the keywords in their content. If nothing is entered in the input field, then all concerts will be listed.

Filter concert by time range

There is an option of filtering the concerts based on time, which makes it easier to find concerts within a specific time period. People can select to show concerts that are displaying within 3 days, a week, a month and so on. The option “show all” will list all concerts.

User

Register/Login

If people want to order a ticket for a concert, they will need to create their account and login as a regular user. They can do this by clicking the login link on the top right of the page and this will direct them to the login page. Once they go to the login page, they will need to register first. They will enter all the information in the register form and it will automatically check if the input values are valid or not. The form can only be submitted if and only if all fields have been filled out and all input values are valid. For example: the username for each user is unique and people cannot use usernames that have been registered by others. Also, people will be asked to enter their password twice and the two passwords have to match.

Once they submit the register form successfully, they will be directed to the login page and they can use their username and password to login. If logged in successfully, they will be directed to the homepage, and the “login” text will be replaced by their full name.

Select seat

Each concert has a total of 26 seats available. Users are allowed to select a seat of a concert by clicking on the “Select Seat” button shown in the bottom of each concert block. Then a pop-up window will come out to show seats that are selectable or occupied, indicated by different colors.

Since each seat is unique for a concert, and each concert corresponds to a total of 26 seats, our ticket inventory for each concert is 26 seats with distinct seat IDs. Once a seat is booked by a user, that is, placed an order and paid by a user, it is not any more selectable by other users.

In our front-end design, the concert hall has a U-shaped layout. Seats are represented by square dots, rendered yellow if they are occupied, green if they are placed in the shopping

cart of the current user but not yet paid, and gray if they are available. If the current user hovers on an available seat, it will turn green, when a tooltip jumps out to show its seat information: column id and row id. When the current user clicks on an available seat, its seat information and price will show in a “Selected Seat” block on the right, which by default is hidden.

Add to Cart

By clicking on “Add To Cart” button in the “Selected Seat” block, the selected seat will be added to the shopping cart. On the backend, we have a table named “suborder” to record each seat added to shopping cart of a user. When a user adds a seat to its shopping cart, a row will be inserted into the table “suborder”. There are three columns for the table: “orderid”, “concertid”, and “seatid”. In our design, each logged-in user is allocated a globally unique integer “orderid” which increments by one for each newly logged in user. When the website is started, an initial integer “orderid” is obtained from the database, by traversing the table “orderinfo” to find the maximum existing “orderid” and plus one. To note, since we considered the “orderid” field to be a string in the database, the maximum “orderid” is actually parsed from string. That is, the automatically incrementing global variable `$GLOBALS['maxOrderid']` is an integer, initially parsed from database, and allocated to a user by setting `$_SESSION['orderid']`.

Check out

Users can place an order by clicking on the “Cart” button in the navigation bar on the top. If the user is not logged in, the “cart” page will jump to the “login” page directly. If the user has not yet added any ticket to the cart, the cart is empty and the “cart” page will jump to the “home” page. If the user is logged in and has added at least one ticket to the cart, the “cart” page will display all tickets placed by the user.

For each ticket, the concert name, the seat row, the seat column, the ticket price, and the concert image will be shown. We also provide three “edit” buttons for each ticket. The “View Seat” button allows the user to view the seat of the ticket. The “Delete” button allows

the user to delete the ticket he/she doesn't want to purchase any more. The "Select" checkbox allows the user to select some tickets in the cart to check out. As long as the user clicks on the checkbox of any ticket, the summary table on the right will calculate and update the total number and the total price of the selected tickets. The user can check or uncheck tickets for unlimited times before deciding to purchase them.

Then the user should go to the summary table and clicks on the "Checkout" button. If no ticket is selected, the page will remind the user to select the tickets. If only some of tickets are selected, the system will checkout the selected ones and remove the unselected ones by assuming the user doesn't want to purchase them any more. These unselected tickets will be available to sell again in our database. After the user successfully place and order, the "successful" page will be shown to remind the state of the order. The user can go the "home" page to select ticket and place another order.

Order history

Users can check order history by clicking on the "My Orders" button in the navigation bar on the top. If the user is not logged in, the "my orders" page will jump to the "login" page directly. If the user has not yet placed any order, the "my orders" page will be empty. Otherwise, the page will display all the orders purchased successfully by the user. For each existing order, we show the order ID, the total price, the pay time, and the detailed information about the tickets (the concert name, the seat row, the seat column and the ticket price) placed in the order.

Logout

If the user wants to log out, he can just click the logout link on the top right of the page. This will log him out and refresh the homepage so his name will be replaced by "login".

Administrator

Add new concert

If logged in as an administrator, the user will be able to add new concert, change the concert information as well as delete an existing concert. After the admin login, there will be an

“Add concert” link on the top right of the page. By clicking it, the admin will be directed to the add concert page. He can enter the information for the concert he wants to add, including name, player, description, price and time. He will also need to upload a picture for the new concert. When uploading the picture, it will check whether the uploaded file is an image and if its size is allowed. Once everything in the add concert form has been filled out, the admin will click submit button and he should be able to see the added concert listed on the home page.

Update concert

The admin can also update the information for an existing concert. By clicking the “edit” button for each concert on the home page, he will be directed to the edit concert page. He can change any information for the concert. If he wants to change the picture, he will need to upload a new picture and again, it will check if the format and size of the image uploaded are allowed. By clicking the submit button, all the changes he made will be saved. There is one exception: if someone has already ordered a ticket for the concert, the admin will not be able to change its price.

Delete concert

To delete an existing concert, the admin will click the “edit” button for the concert he wants to delete and he will be directed to the edit concert page. On the top right of the page, there is a delete link. The admin can just click this link to delete the concert. However, if someone has ordered tickets for the concert, the admin will not be allowed to delete it.

Project Assumption

- A user can add only one ticket to the cart every time.
- We will remove unselected tickets in the cart when you check out.
- One concert has only one show time option.
- All concerts are held in the same theater.
- Users cannot add the tickets to the cart if the tickets are reserved or purchased by others.

- The admin can only delete or change price for the concert that no one has ordered ticket for. Soft delete was implemented for deleting concert, so the information for the deleted concert is still available in the database.

Database Design

```
-- phpMyAdmin SQL Dump
-- version 4.6.5.2
-- https://www.phpmyadmin.net/
--
-- Host: localhost:8889
-- Generation Time: Aug 08, 2017 at 04:39 AM
-- Server version: 5.6.35
-- PHP Version: 7.0.15

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

-- Database: `Project`
--

-----
-- Table structure for table `concert`
--

CREATE TABLE `concert` (
  `concertid` varchar(30) NOT NULL,
  `concertname` varchar(30) NOT NULL,
  `time` datetime NOT NULL,
  `player` varchar(30) NOT NULL,
  `description` varchar(500) NOT NULL,
  `image` varchar(30) NOT NULL,
  `price` int(11) NOT NULL,
  `isshown` tinyint(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
```

```
-- Table structure for table `orderinfo`
-- 

CREATE TABLE `orderinfo` (
  `orderid` varchar(30) NOT NULL,
  `userid` varchar(30) NOT NULL,
  `paytime` datetime NOT NULL,
  `status` tinyint(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- 
-- Table structure for table `seat`
```

```
-- 
CREATE TABLE `seat` (
  `seatid` int(11) NOT NULL,
  `row` int(11) NOT NULL,
  `col` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- 
-- Table structure for table `suborder`
```

```
-- 
CREATE TABLE `suborder` (
  `orderid` varchar(30) NOT NULL,
  `concertid` varchar(30) NOT NULL,
  `seatid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- 
-- Table structure for table `users`
```

```
-- 
CREATE TABLE `users` (
  `userid` varchar(30) NOT NULL,
  `firstname` varchar(30) NOT NULL,
  `lastname` varchar(30) NOT NULL,
```

```

`password` varchar(64) NOT NULL,
`email` varchar(30) NOT NULL,
`authority` varchar(30) NOT NULL,
`salt` varchar(64) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `concert`
-- 
ALTER TABLE `concert`
ADD PRIMARY KEY (`concertid`);

-- 
-- Indexes for table `orderinfo`
-- 
ALTER TABLE `orderinfo`
ADD PRIMARY KEY (`orderid`),
ADD KEY `rf1` (`userid`);

-- 
-- Indexes for table `seat`
-- 
ALTER TABLE `seat`
ADD PRIMARY KEY (`seatid`);

-- 
-- Indexes for table `suborder`
-- 
ALTER TABLE `suborder`
ADD PRIMARY KEY (`orderid`,`concertid`,`seatid`),
ADD KEY `rf3` (`seatid`),
ADD KEY `rf4` (`concertid`);

-- 
-- Indexes for table `users`
-- 
ALTER TABLE `users`
ADD PRIMARY KEY (`userid`);

-- 

```

```
-- Constraints for dumped tables
--
-- Constraints for table `orderinfo`
--
ALTER TABLE `orderinfo`
  ADD CONSTRAINT `rf1` FOREIGN KEY (`userid`) REFERENCES `users`(`userid`);

--
-- Constraints for table `suborder`
--
ALTER TABLE `suborder`
  ADD CONSTRAINT `rf2` FOREIGN KEY (`orderid`) REFERENCES `orderinfo`(`orderid`),
  ADD CONSTRAINT `rf3` FOREIGN KEY (`seatid`) REFERENCES `seat`(`seatid`),
  ADD CONSTRAINT `rf4` FOREIGN KEY (`concertid`) REFERENCES `orderinfo`(`orderid');
```

Language/Framework

Client-side: HTML, CSS, JavaScript, jQuery, AJAX

Server-side: PHP

Back-end: MySQL

Frameworks: Bootstrap

Screenshots

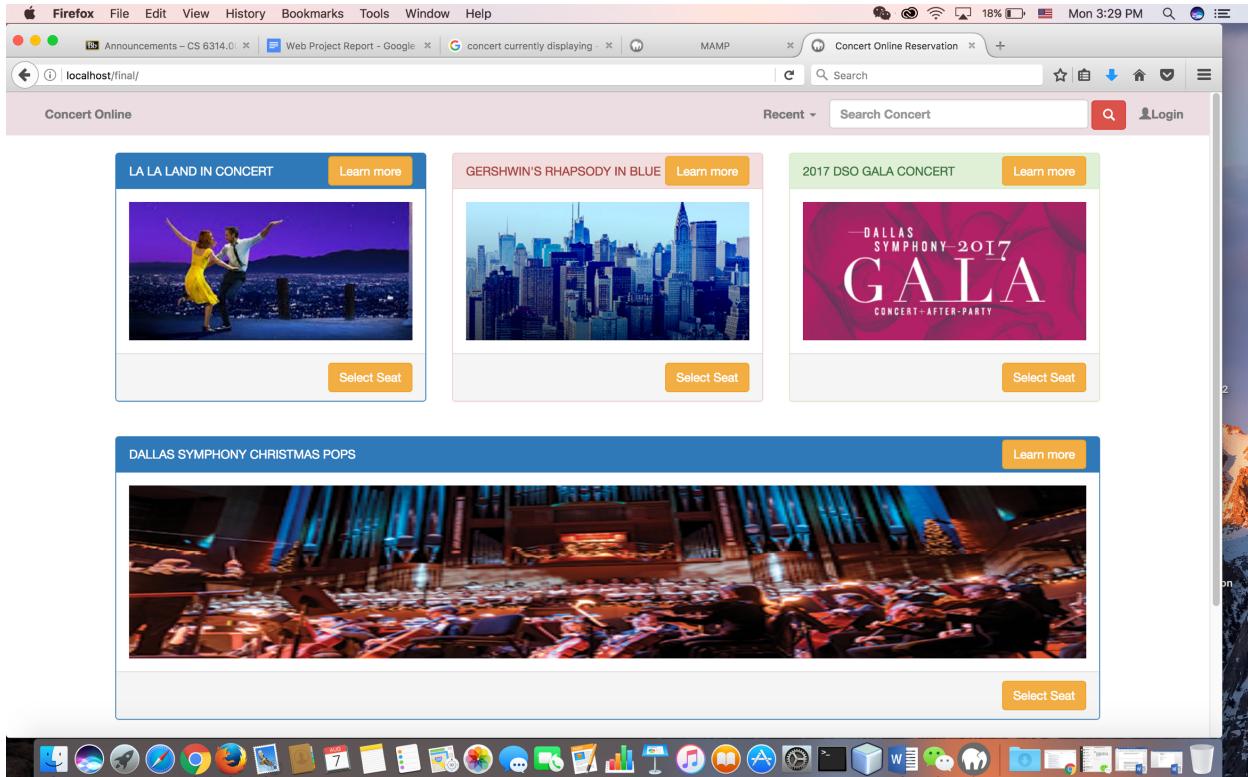


Figure 1 Home page

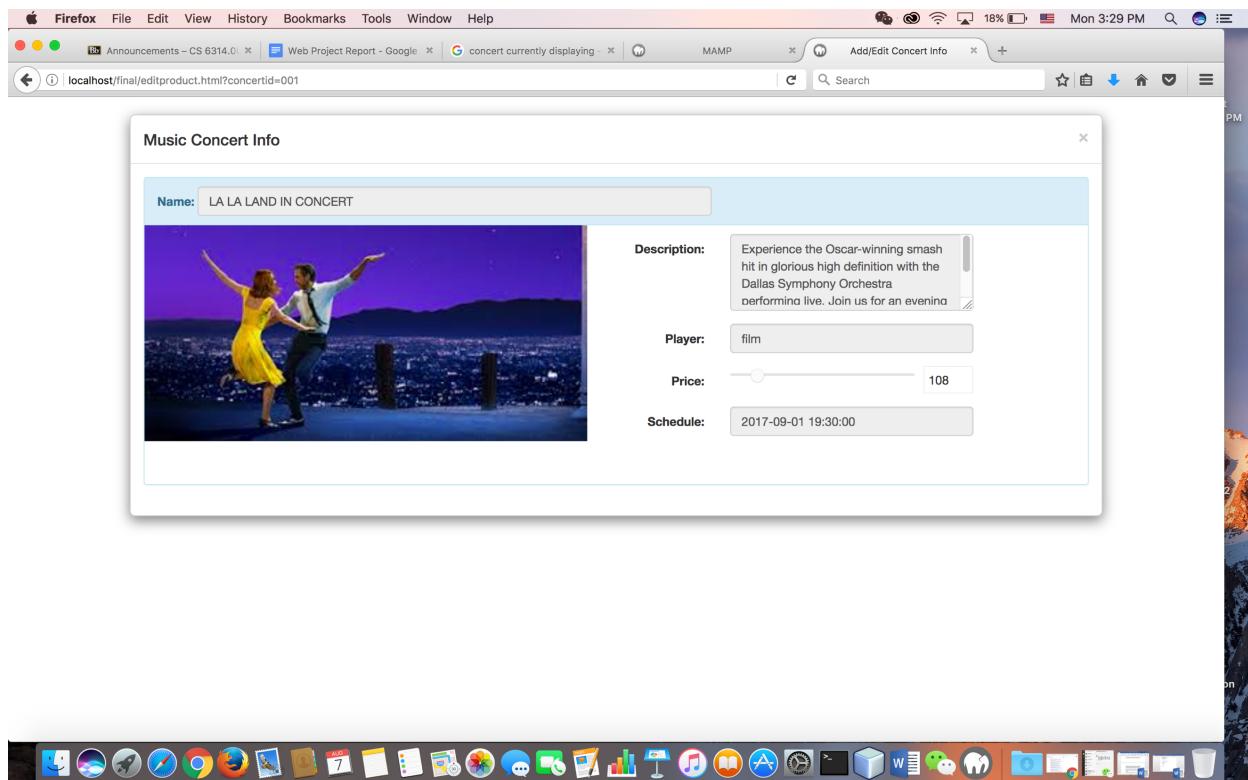


Figure 2 Concert information page

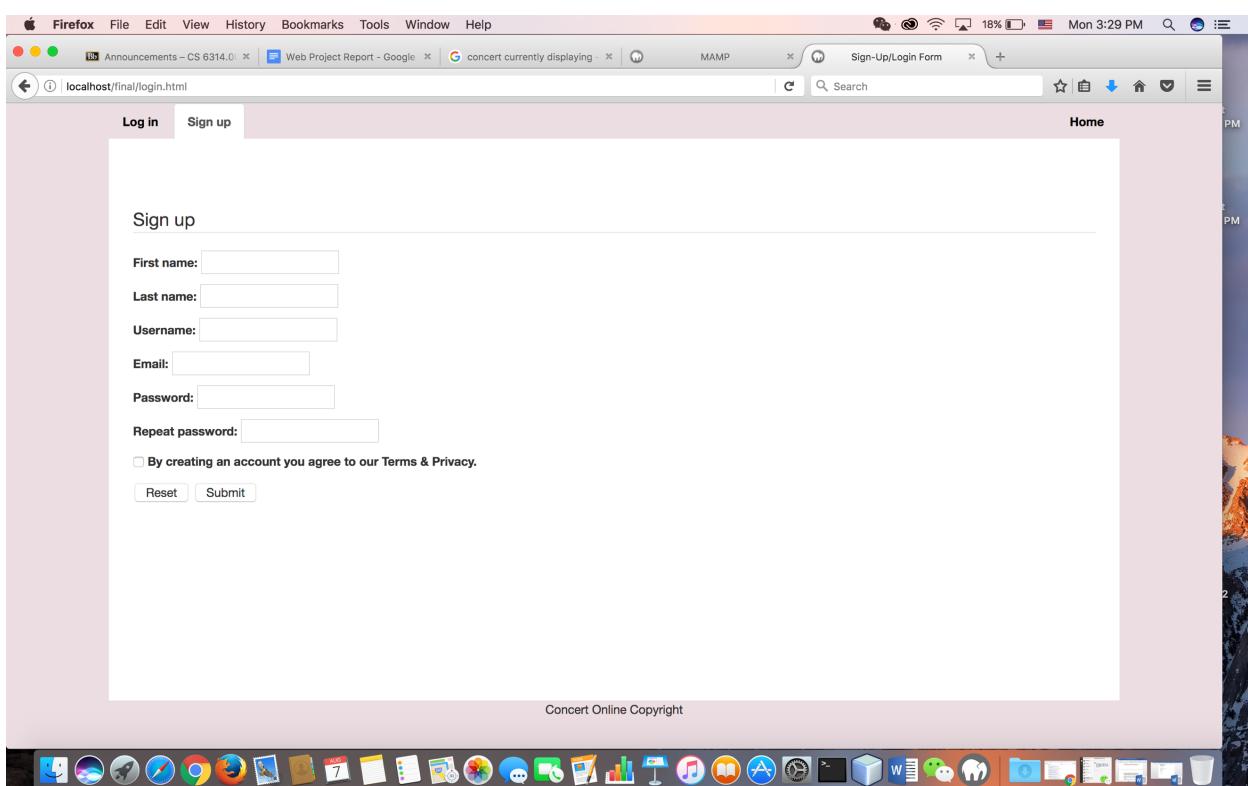


Figure 3 Register page

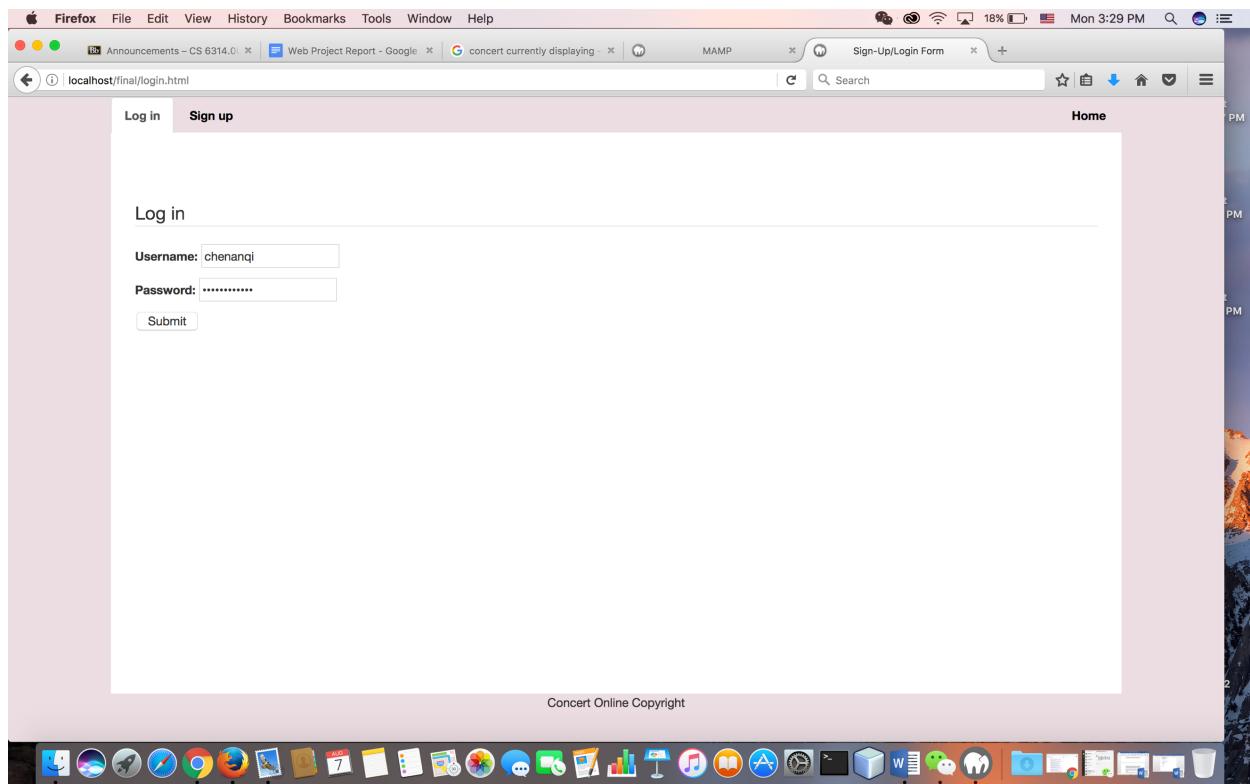


Figure 4 Login page

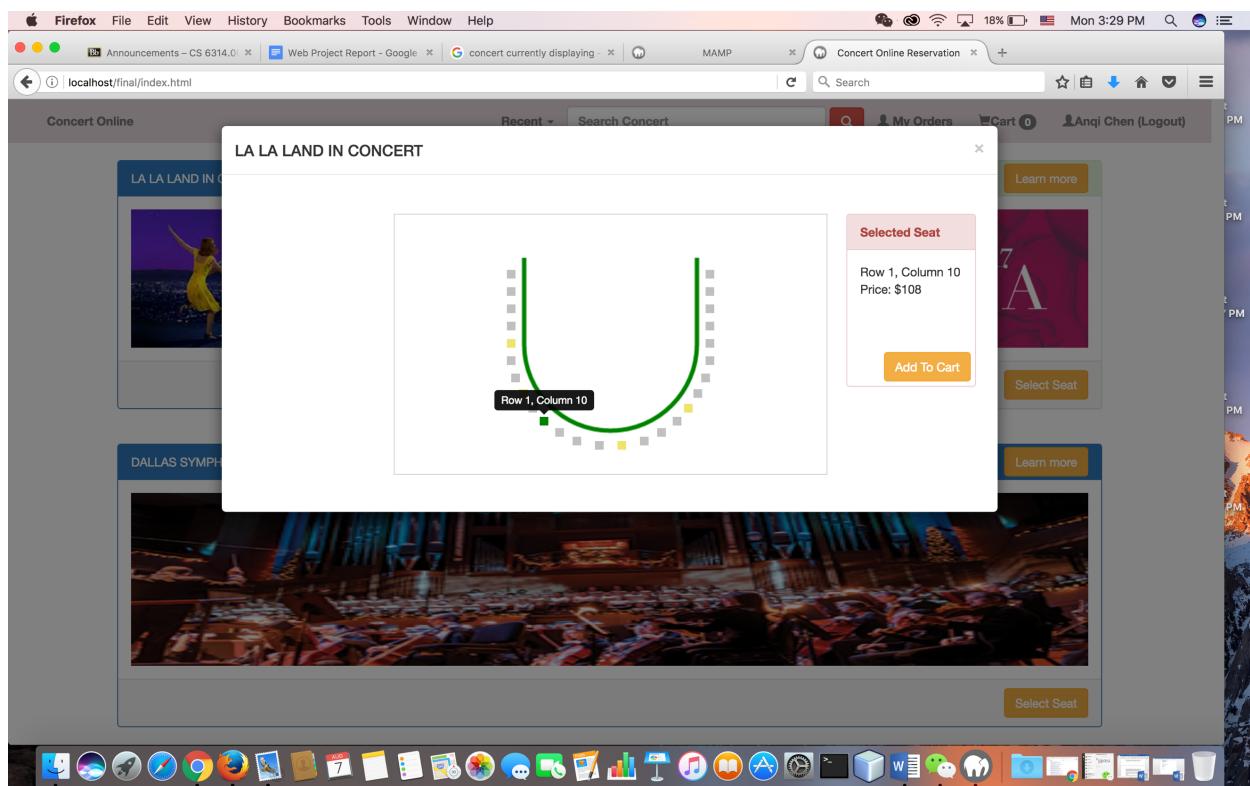


Figure 5 Select seat page

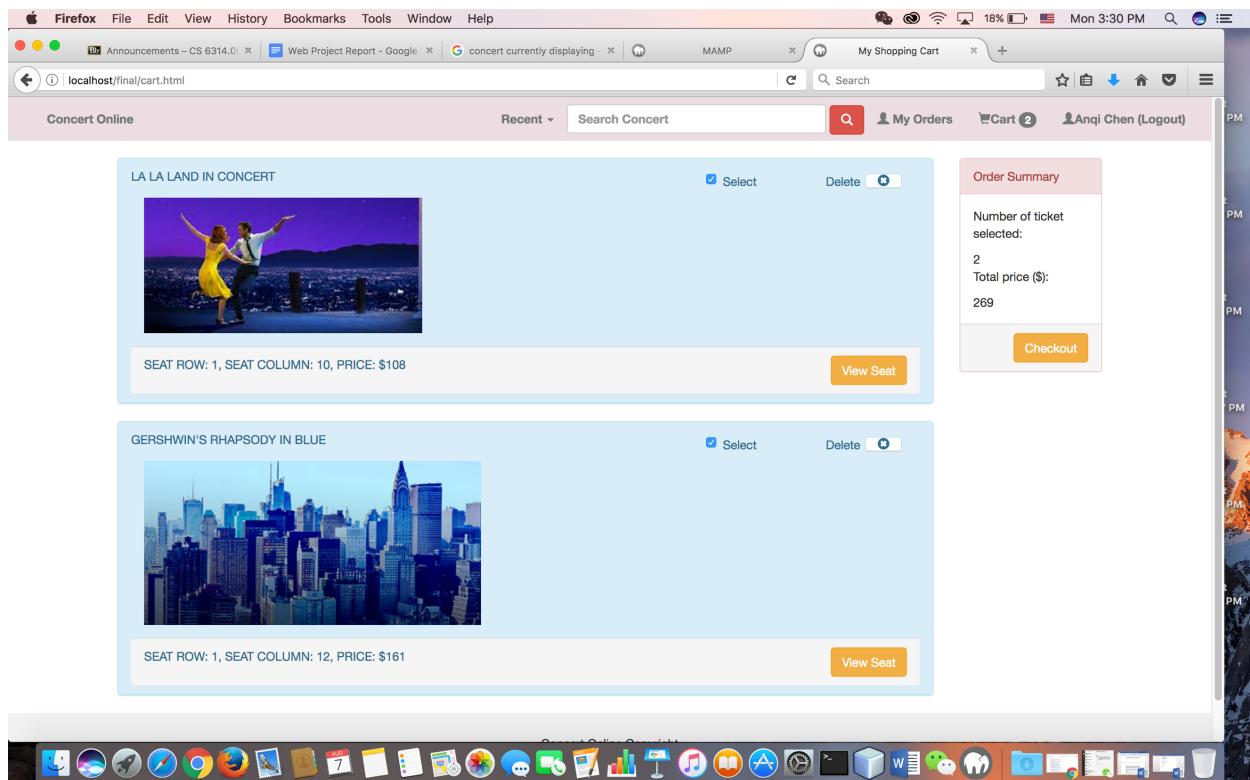


Figure 6 Shopping cart

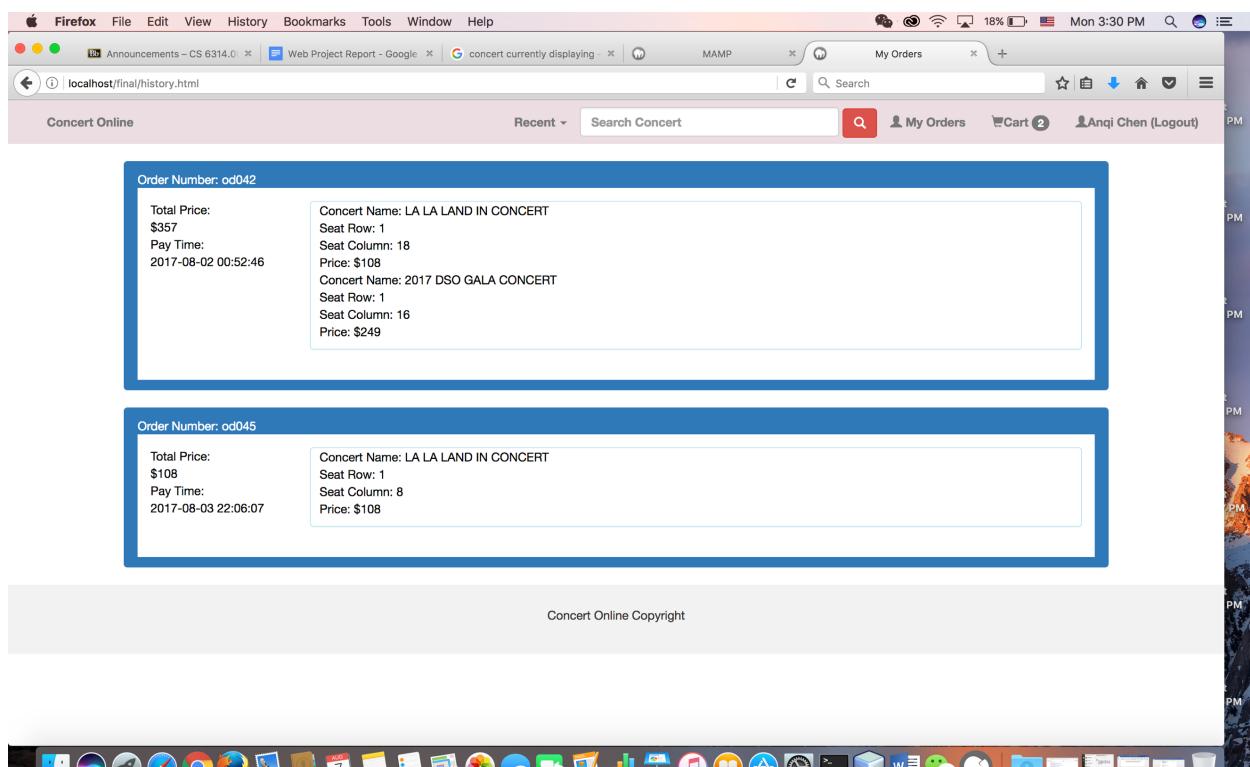


Figure 7 Order history

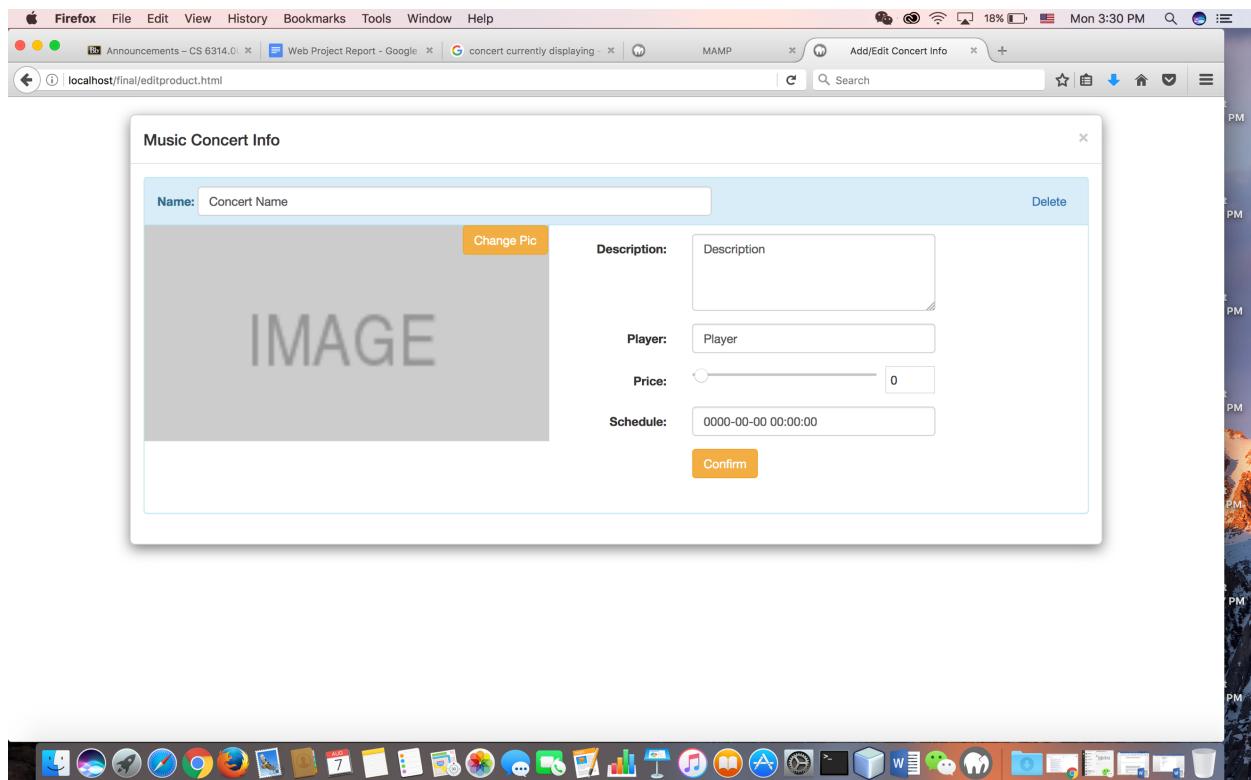


Figure 8 Add new concert

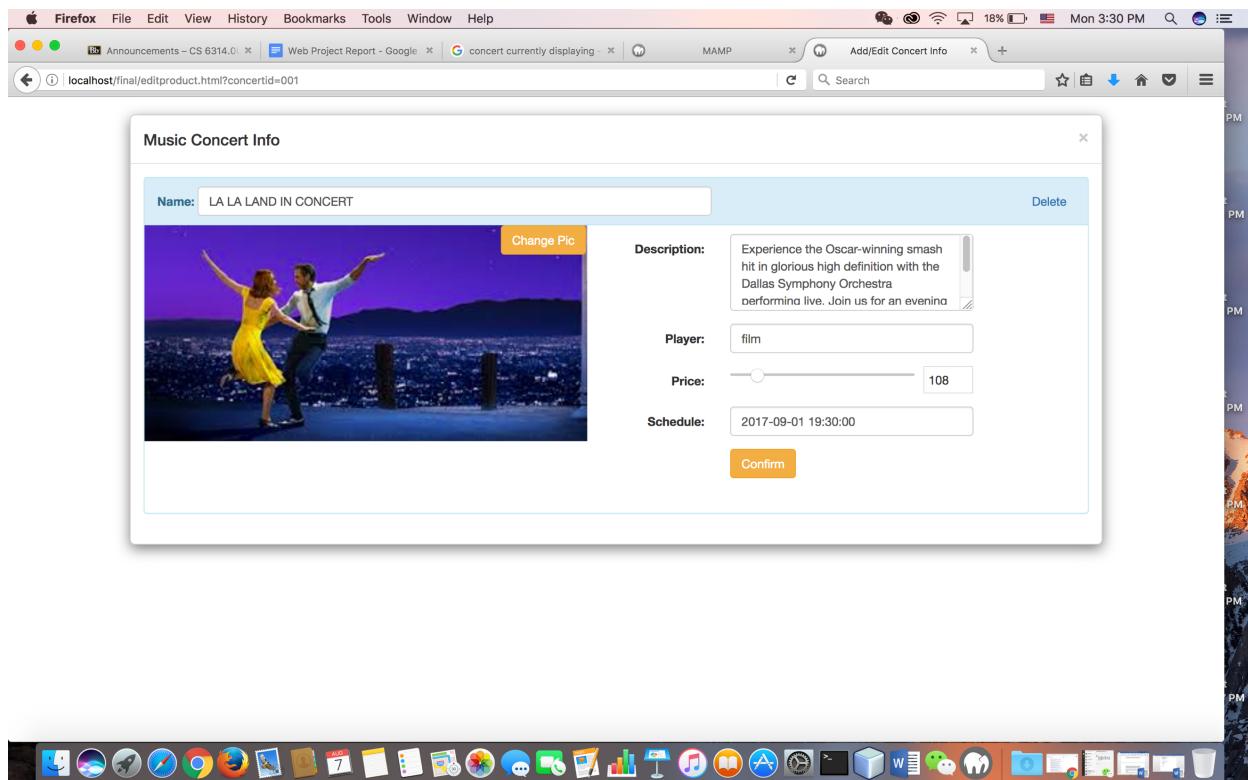


Figure 9 Edit concert page

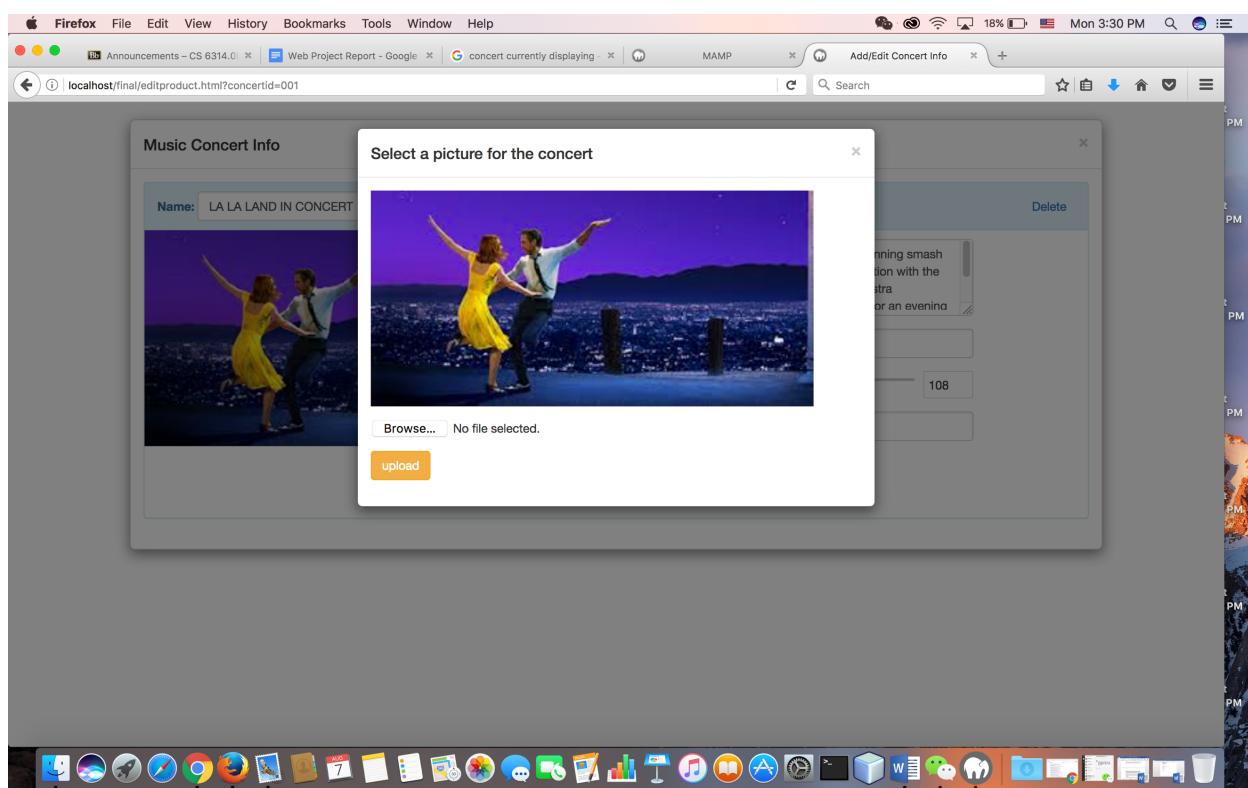


Figure 10 Upload image page

Work Division

Anqi Chen

1. Worked on the front-end, back-end design, client side and server side scripting of the login/signup page.
2. Worked on the client side and server side scripting for adding concert, editing concert and deleting concert.
3. Worked on the client side and server side scripting for showing concert on home page, including creating pagination, as well as searching concert based on keywords and filtering concert based on time.

Qianying Lin

1. Designed and implemented the database.
2. Implement functions in the cart page: alert about no logging, alter about no item in cart, dynamically show the suborder list, delete order and reload page, select order and show total price in summary table, alter no selection when check out, check out selected order and remove unselected suborder, check out and change pay time, after check out increase the orderid, then goes to the successful.html
3. Implement functions in order history page: dynamically show order history, alert when no history, alert when no logging
4. Edit navigation bar to keep showing login status in cart.html and history.html
5. Implement the dynamically updating the number of items in the cart

Juanjuan Zhao

1. Coded the front-end of most html pages using Bootstrap, including the main page, shopping cart list page, order list page, administrator's product editing page, seat

selection popup window page. The seating chart is drawn dynamically using html5 canvas.

2. Realized the back-end of seat selection module: add a selected seat by inserting a row in table “suborder”; show occupied or available seats on the seating chart by reading the table “suborder”.
3. Solved the issue of how to globally uniquely allocate an orderid to a user, by initially reading existing maximum orderid from database, and subsequently increment global variable \$GLOBALS['maxOrderid'] by one for each login action.
4. Fixed bugs in front-end effects to increase user-friendliness after major functional modules are completed by other team members, including page jumping issues, buttons flashing in the navigation when a user logs in or off.