**Camping Reservation**

**Name of the website:** Camping Reservation System

**Project Description**

There are two different user roles assumed in using the website, one being the administrator and the other being the end-user (customer). The administrator is a superuser who has extra privileges of adding/deleting/reading/updating all the camping information. Customers are the end-users who use the system for making a reservation.

The Camping Information can only be viewed by authenticated users. You need to have a valid user name and password. A new user can go to the Register Page and create his user credentials. Username and email are unique fields, so two users cannot be registered using the same email id or user name. The password entered by the user is encrypted, so it cannot be viewed even by the administrator.

Once a user is logged into the website. He will be directed to the main page which show the information of all the camps. Pagination is implemented here. So only two camps can one shown at one time. User can also search for a specific camp by typing in the keywords in the Search Criteria textbox. User can also filter his search based on city, state or camp name. A user can also book a camp by clicking on the Book button next to a camp. User is then directed to book camp page. Here he/she must select the reservation check in and check out dates along with the number of slots that he/she wants to reserve. Reserve slots page is first disabled. User has click on Check Available slots button after selecting the required field. Reserve Slots button is only enabled if slots are available. After clicking on Reserve Slots button, user will be directed to the payment page. Here the user must enter his credit/debit card details. Payment functionality is not validated in our website. It is assumed that the user enters correct details. After the payment is made, the user will get an alert saying that Reservation is Successful and he/she will be redirected to the mail page.

A user can go to My Reservations to check all the reservations that he has made. User can cancel a future reservation. If a reservation is started, then it cannot be cancelled by the user.

Admin user has the same interface as the end user, except that it has an extra feature of editing a reservation. An extra Edit Camp tab is seen in the admin user’s interface. Admin user can you to Edit Camp. Here he/she can add a new camp, modify an existing camp or delete any camp. We have implemented a soft delete. Delete flag will be set if the administrator deleted any camp. The information regarding that camp will still be accessible in the database.

**Database Design**

Camping Site table stores all the information regarding a camp.

CAMPING\_ID- Unique id associated with each camp. It is an auto increment field.

NAME- Name of the camp

DESC\_ID- Foreign key which reference the Description table

STREET- Street address of the camp

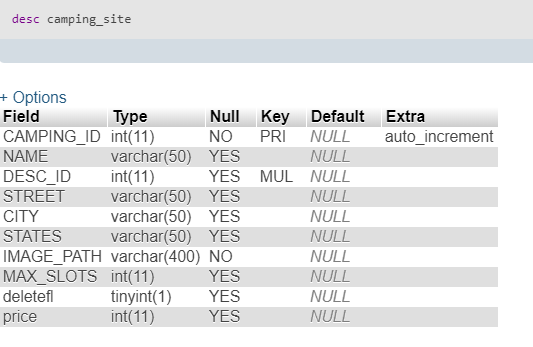
CITY- City name

STATES- State name

IMAGE\_PATH- Image path

deletefl- Delete flag

price- price of each slot in the camp



Description table contains all the descriptive information about the camp.

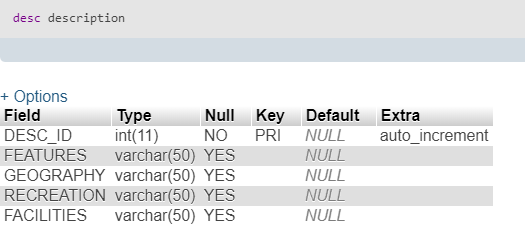
DESC\_ID- Unique description id associated with each camp. It is an autoincrement field.

FEATURES- Features of the camp

GEOGRAPHY- Geography of the camp

RECREATION- Recreation activities

FACILITIES- Facilities offer by the camp



Reserved slots table contains all the Reservation Information.

RESERVATION\_ID- Unique reservation id associated with each reservation. It is an autoincrement field.

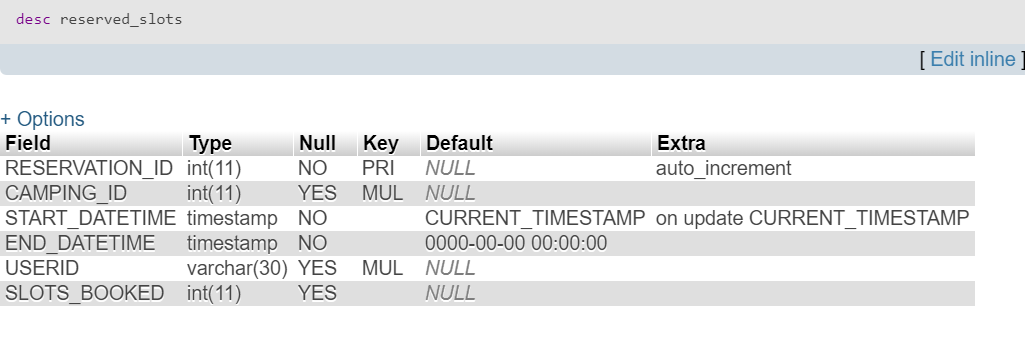
CAMPING\_ID- Foreign key which refers the camping table.

START\_DATETIME- Starting date of the reservation

END\_DATETIME- End date of the reservation

USERID- Foreign key which references the users table

SLOTS\_BOOKED- The Number of slots booked by each reservation

 Users table contains the information regarding each user.

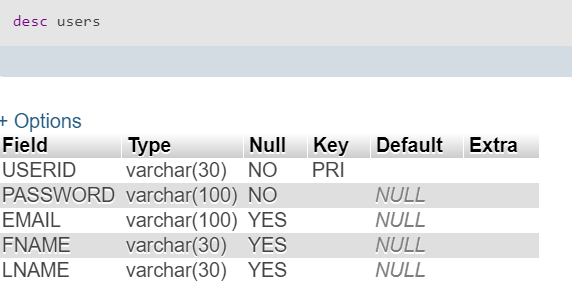
USERID- Unique userid of the user.

PASSWORD- Password of the user

EMAIL- Email of the user

FNAME- First name of the user

LNAME- Last name of the user



The password that is stored in DB is MD5 hashed. The PHP script does internally the MD5 hashing and password verification during user login and user registration automatically.



The SQL file attached to the submission contains all the required statements to set up the DB.

**Languages/Frameworks used and other technical requirements**

|  |  |
| --- | --- |
| Front-end | JQuery (incl. AJAX Calls to PHP), HTML, Bootstrap |
| Back-end | PHP (AJAX response thru. JSON, as applicable) |
| Database | MySQL Community Edition 5.7 |
| OS Developed | Windows 10 |
| Platform supported | Windows, Linux, MAC |
| Pre-requisites | Apache Web server, MySQL (Root user credentials – root/root) |
| Browsers tested | Google Chrome |

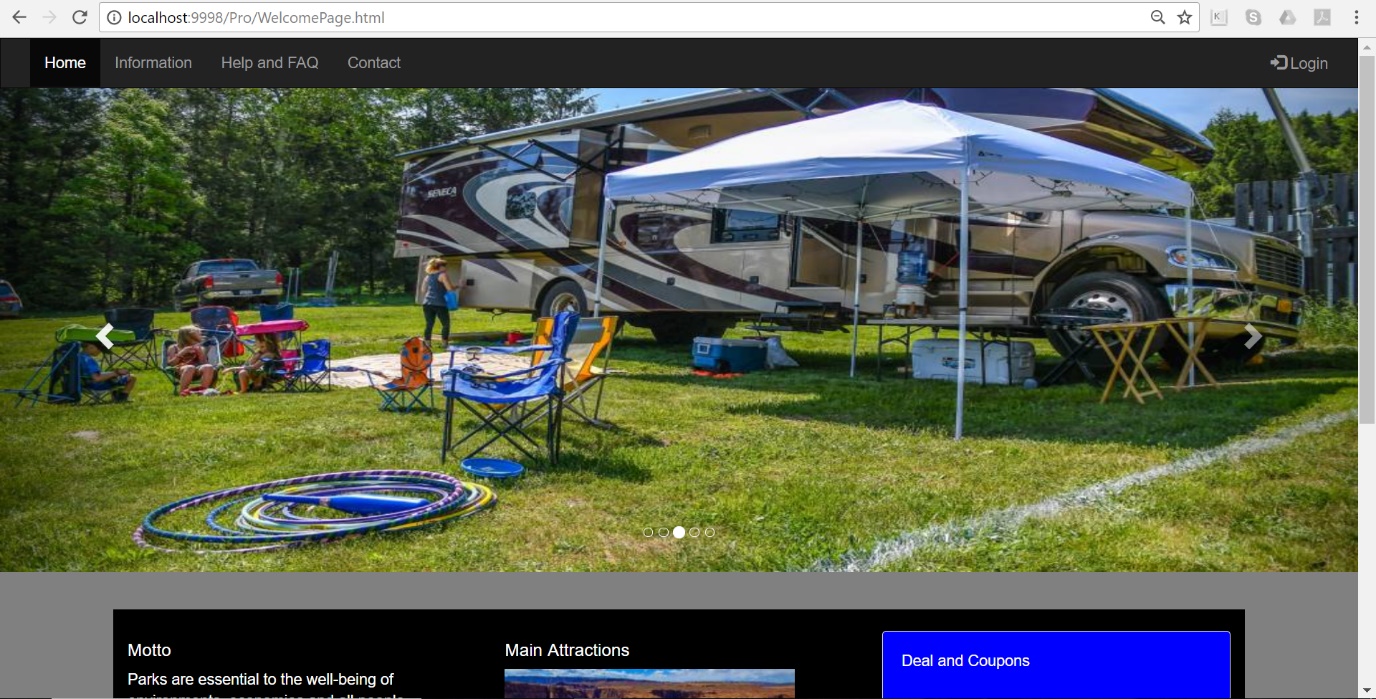
In case if MySQL root user credential is to be changed, then the same needs to be updated in the back-end PHP scripts as well.

Also, it is expected that the target environment meets all the pre-requisites and the source code is put under the Apache web service folder for the web-service to serve the website.

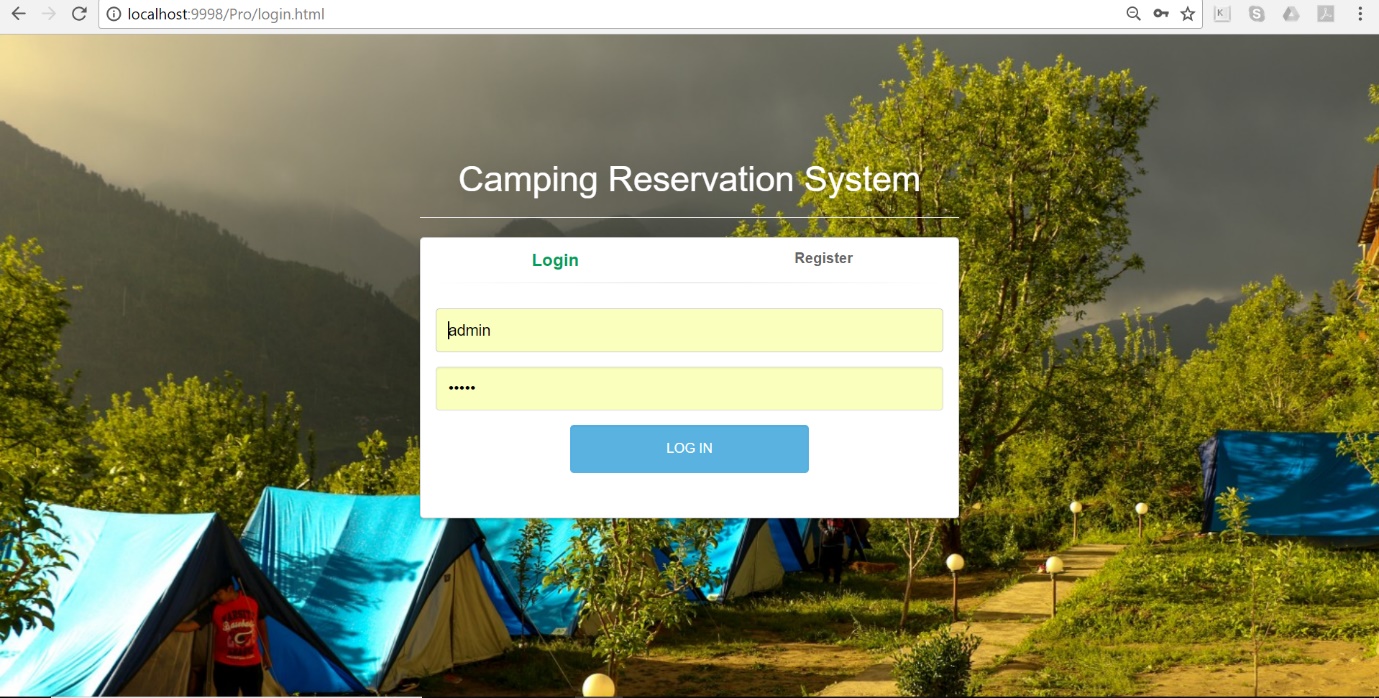
**Screen-shots**

**End User related:**

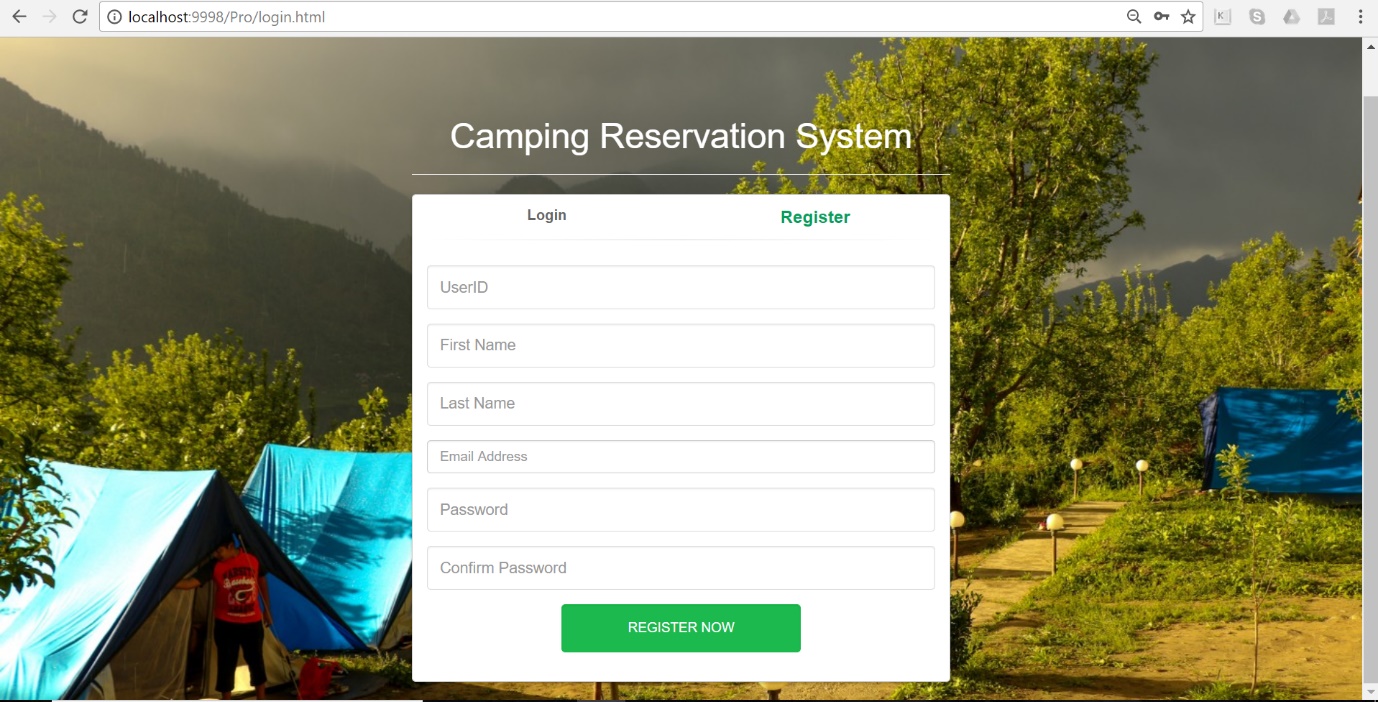
**Welcome Page**



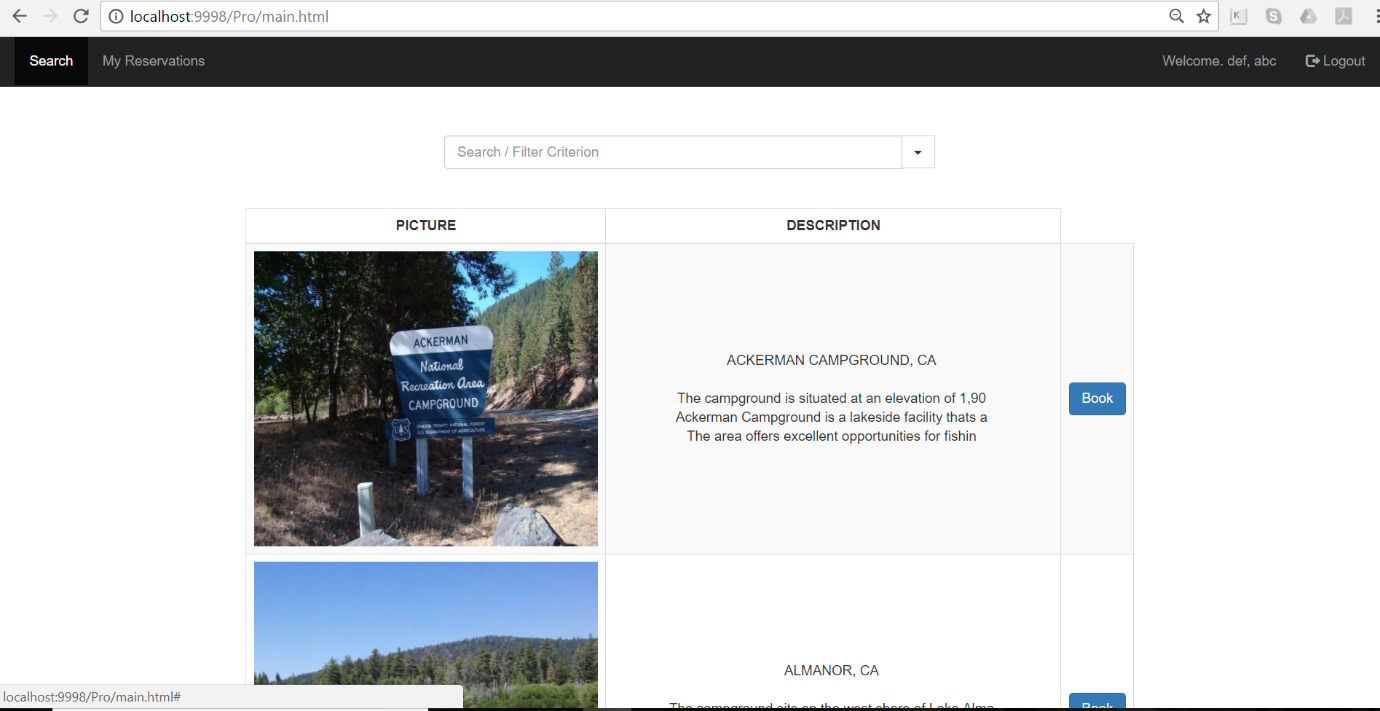
**Login Page**



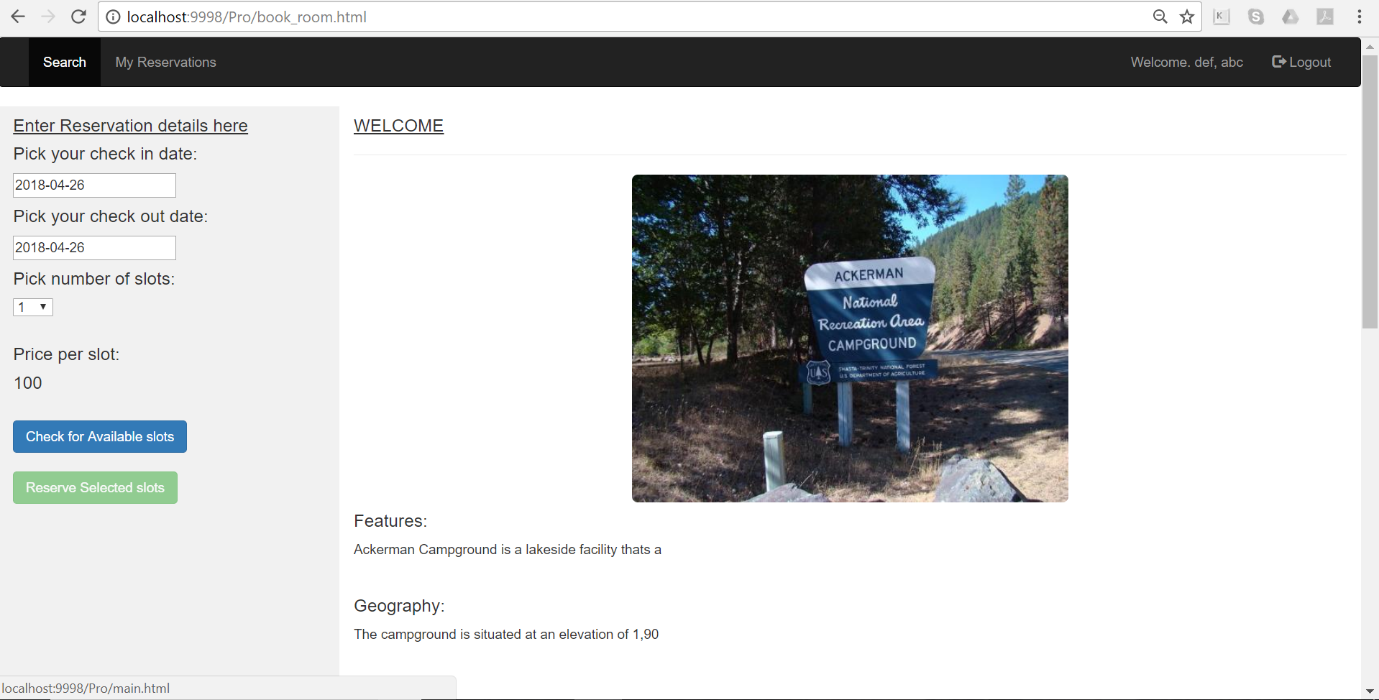
**Register Page**



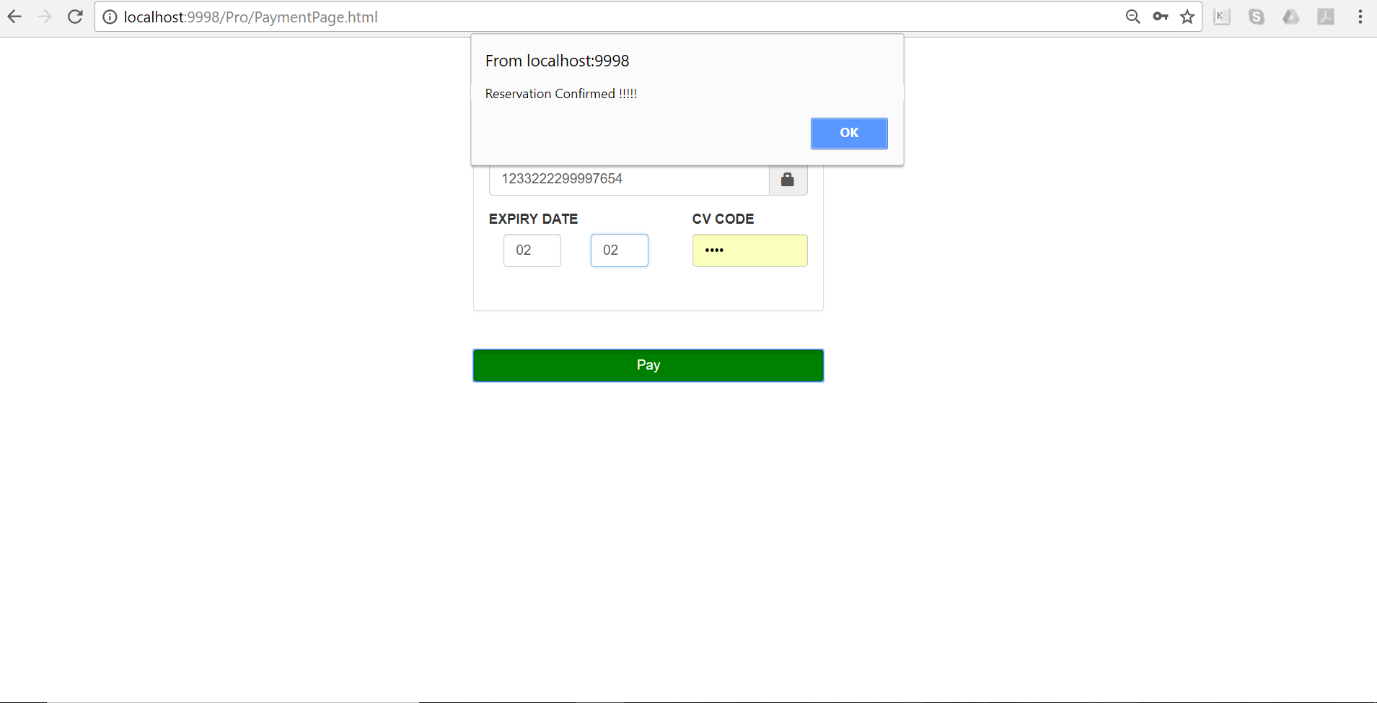
**Main Page**



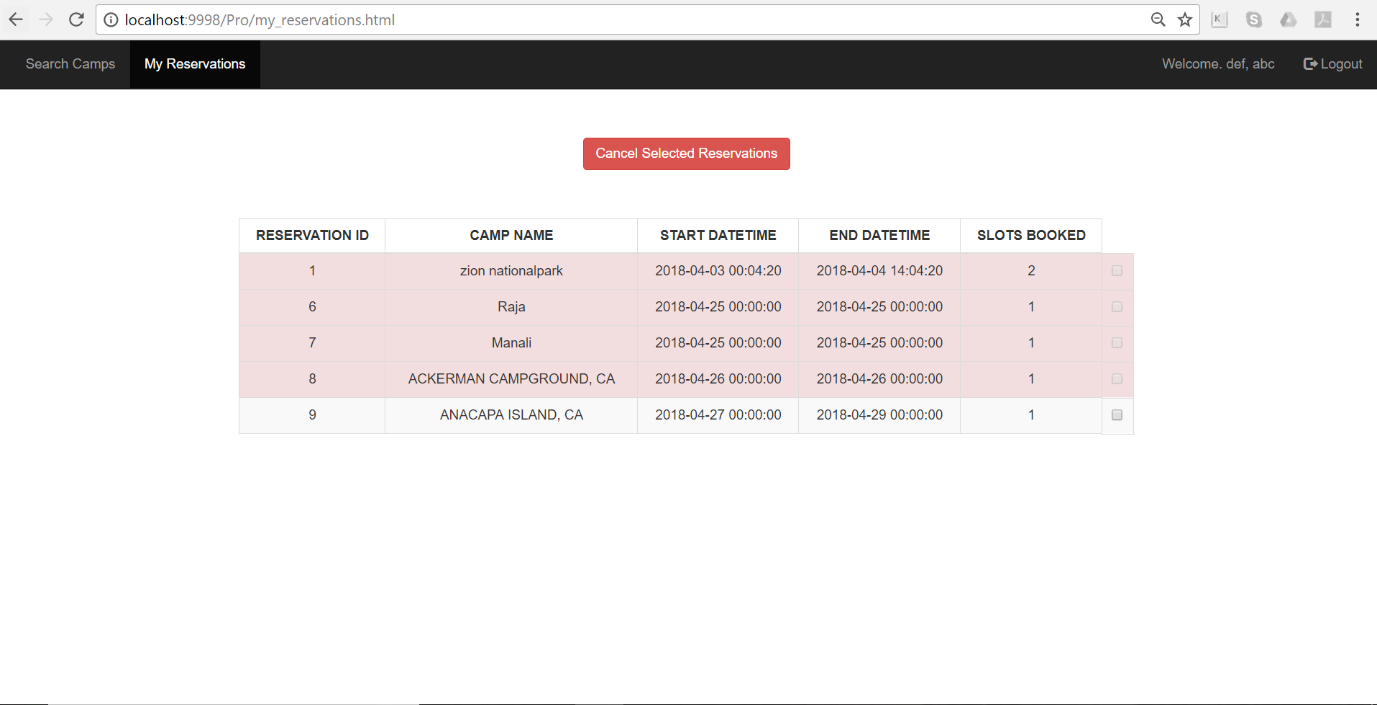
**Book a Camp**



**Payment Page with Reservations Confirmation**

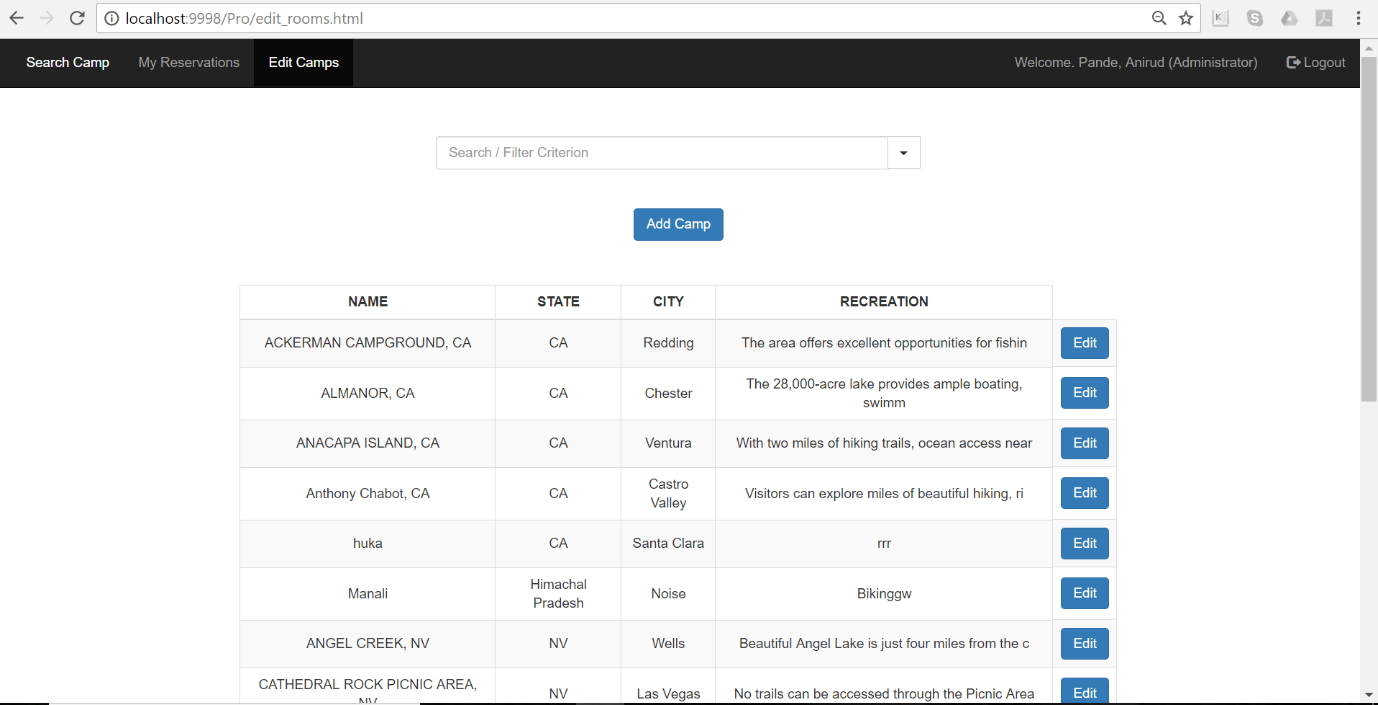


**My Reservations Page**

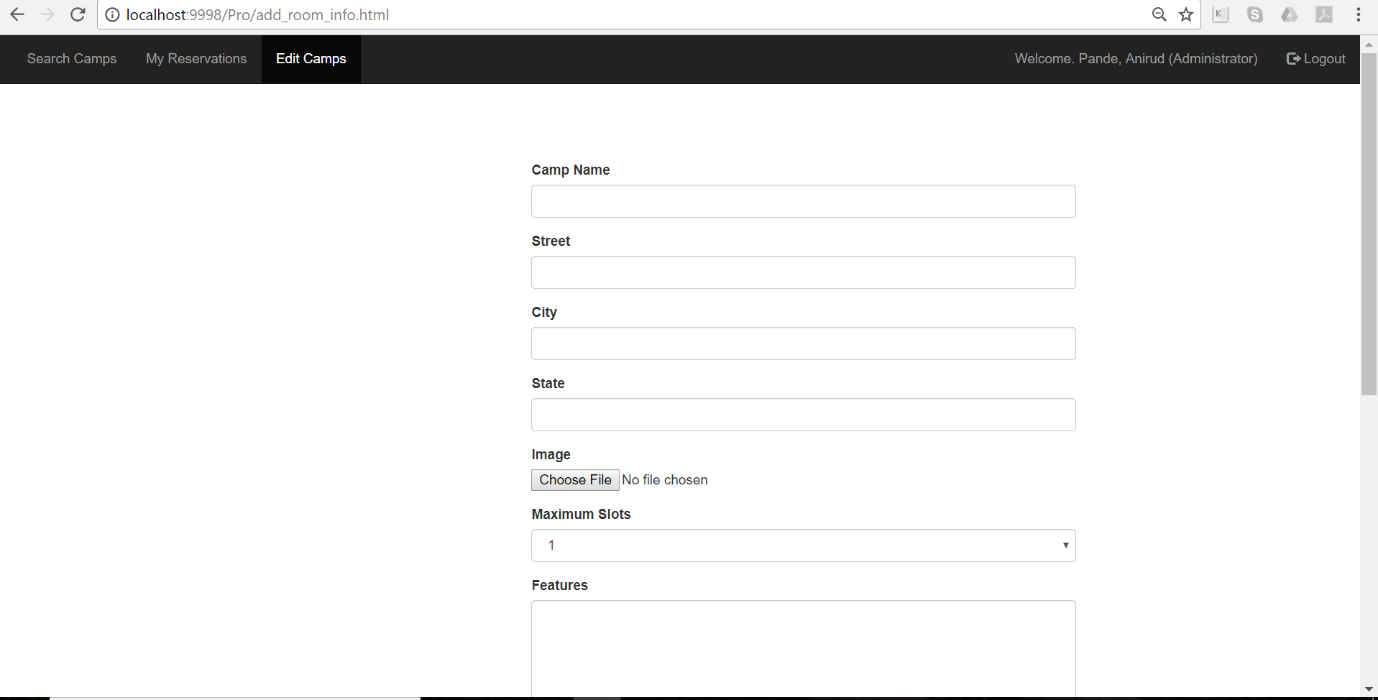


**Administrator related:**

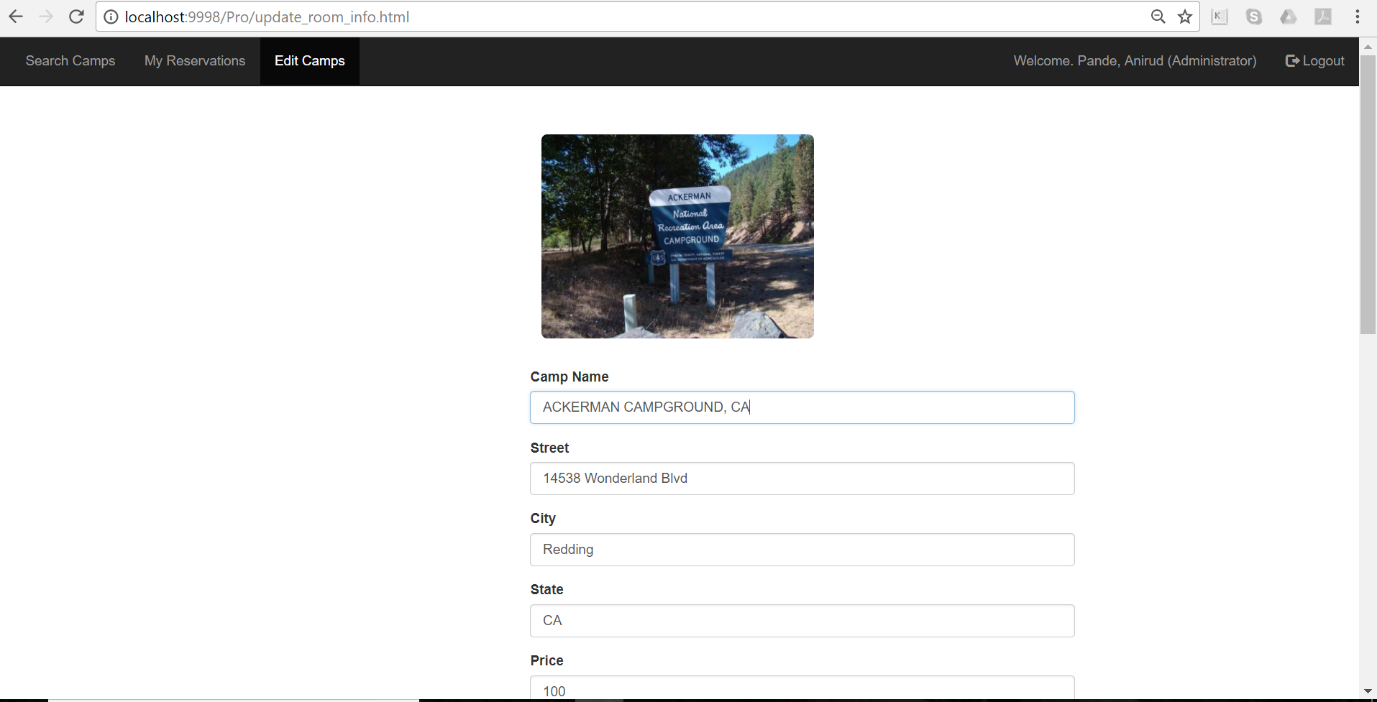
**Edit Reservations Page**



**Add New Camp**



**Edit existing camp information**



**Search Box in Main Page**

