Implementation and Testing

Project titile: Online Food Ordering & Reservation System
Group Name: Group 9
Members: Mingwei Sui, Zhikai Lin, Hongliang Gao

一. Execution Instructions

[10 points]: detailed instruction on how to run and execute your project. Make sure to include all the required information to successfully execute your project (e.g., OS, Libraries, etc.)

1. Minimum Requirement to run our project:

OS/Software Libraries/Flask extensions	Description
Linux Operating system/Mac OS X	The recommended production deployment environment is lastest Linux Operating system including Debian, CentOS, Ubuntu.
Python 2.7.x	To ensure the better compatibility of different packages involved in our project. We do not recommend to use python 3.x as a software environment since there are still a lot of python packages that do not well cooperate with each other.
Flask 1.0.2	The python web framework, which is MVC-based microframework compared with Django framework. The community provides flask with a large set of extensions that allow us to build a full-featured MVC Framework.
bcrypt/Flask-Bcrypt 0.7.1	To ensure the security of sensitive data stored in database. We use this password hashing extension, which provides us a feature encrypting our password stored in the database.
Flask-Bootstrap 3.3.7.1	We use flask-boostrap, a integration of flask with one of the most popular front-end framework, bootstrap. Therefor significantly reducing the total workload of UI development in our project.
Flask-SQLAlchemy/SQLAlchemy	Flask application use SQLAlchemy as a ORM framework component/SQL toolkit.

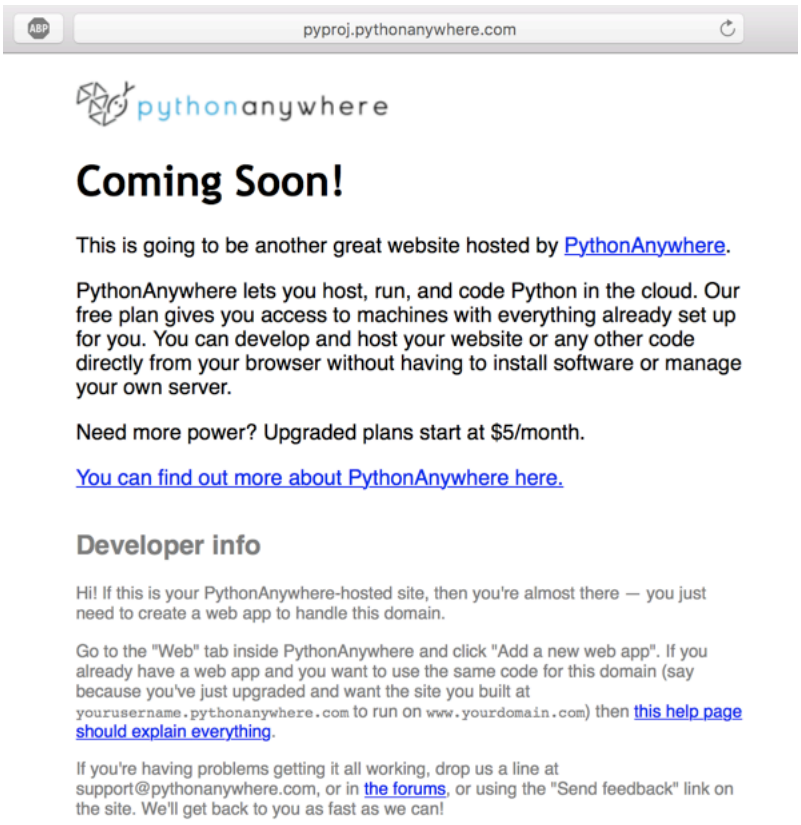
Flask-Login	This extension of flask helps us manage user session including logging in, logging out and etc.
Flask-Migrate	This extension handles SQLAlchemy database migrations for flask application. It is very helpful when we make change to the database during development including changing the column name of a column, type of a column and size of a field and adding or deleting a column and etc.
Flask-WTF/WTForms	WTForms is a flexible forms validation and rendering library for Python web development.
Jinja2	Jinja2 is a full featured template engine for Python.
MYSQL-python	This extension provides an interface to the popular MYSQL database server for python
boostrap4.1	front-end html framework
boostrap datetimepicker	datetime selector component
Jquery	javascript library required to supoort other component above.

2. installation:

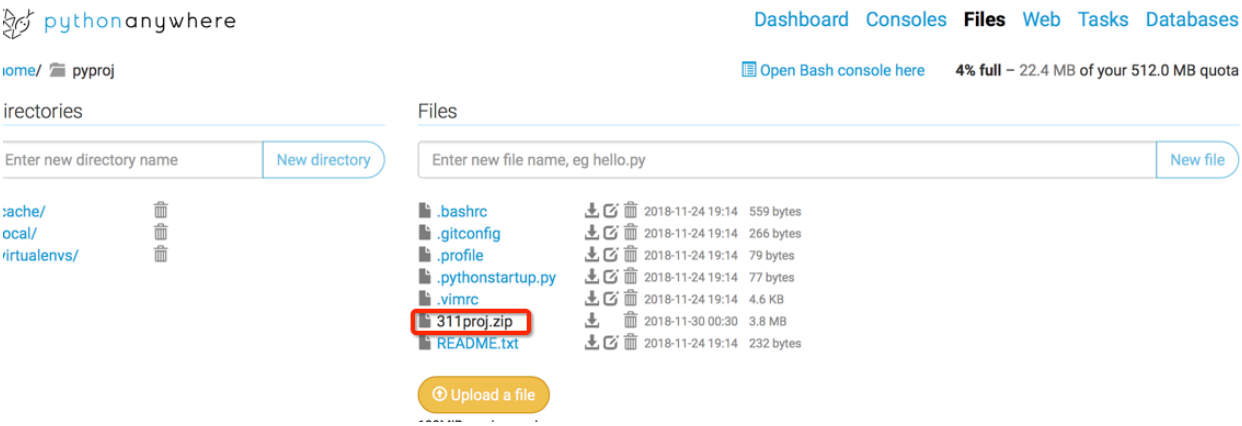
To make it simple, we demonstrate the installation and running our project on the PythonAnywhere. PythonAnywhere is an online integrated development environment and web hosting service based on the Python programming language.

(1) PythonAnywhere Set-Up

Create a free PythonAnywhere account here. The usename will be used to as a domain of access URL of the project. The access URL will be **http://yourusername.pythonanywhere.com**. We create a username pyproj, which allows us to access our project by accessing http://pyproj.pythonanywhere.com, the figure below shows the default page after an account created.



(2). upload our project files



unzip project zip file:

execute command**:** unzip 311proj.zip

```
(fors) 00:51 ~ $ unzip 311proj.zip
```

(3).Python virtual environment

in bash console:

execute command: $mkvirtualenv fors, "fors" is a name for virtual python environment created to running our project.

```
00:18 ~ $ mkvirtualenv  fors
New python executable in /home/pyproj/.virtualenvs/fors/bin/python2.7
Also creating executable in /home/pyproj/.virtualenvs/fors/bin/python
Installing setuptools, pip, wheel...
done.
virtualenvwrapper.user_scripts creating /home/pyproj/.virtualenvs/fors/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/pyproj/.virtualenvs/fors/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/pyproj/.virtualenvs/fors/bin/preactivate
virtualenvwrapper.user_scripts creating /home/pyproj/.virtualenvs/fors/bin/postactivate
virtualenvwrapper.user_scripts creating /home/pyproj/.virtualenvs/fors/bin/get_env_details
(fors) 00:26 ~ $
(fors) 00:26 ~ $
```

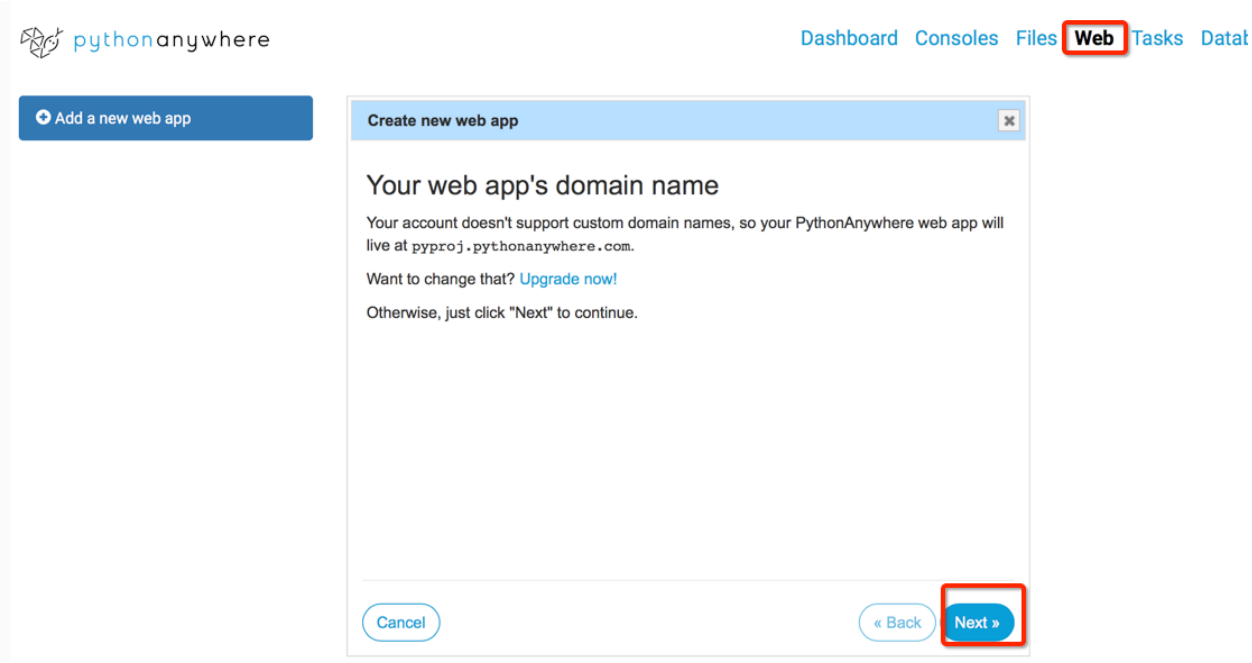
(4) install all flask extensions required by the project, execute command:

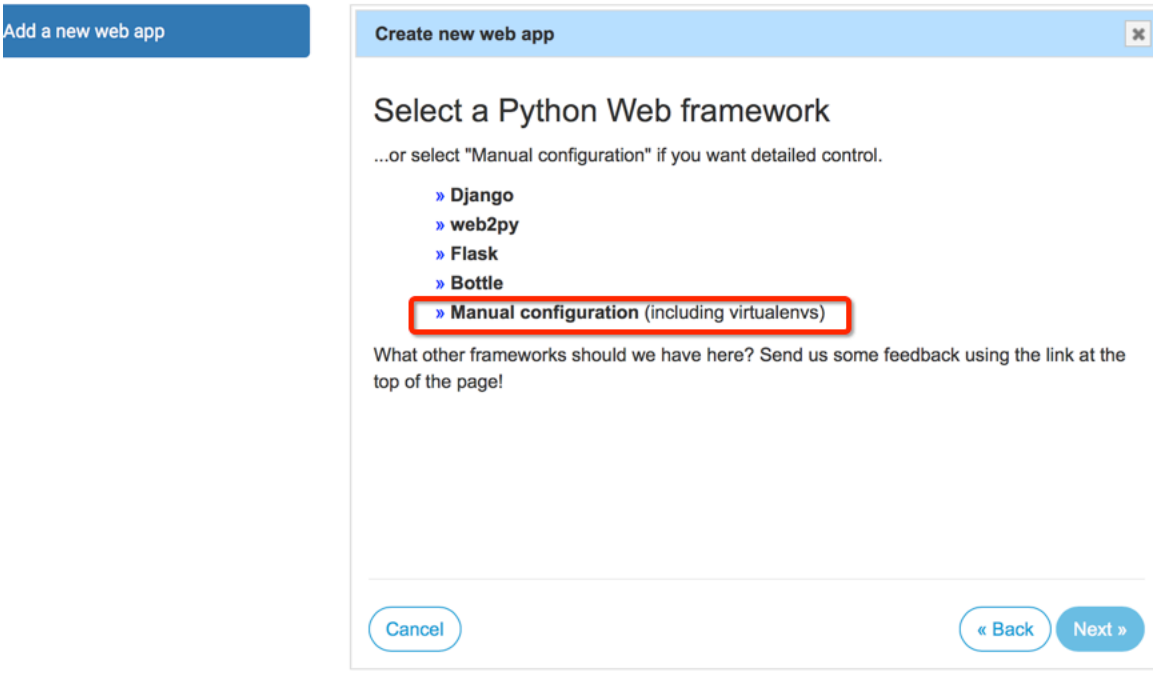
```
pip install -r requirements.txt
```

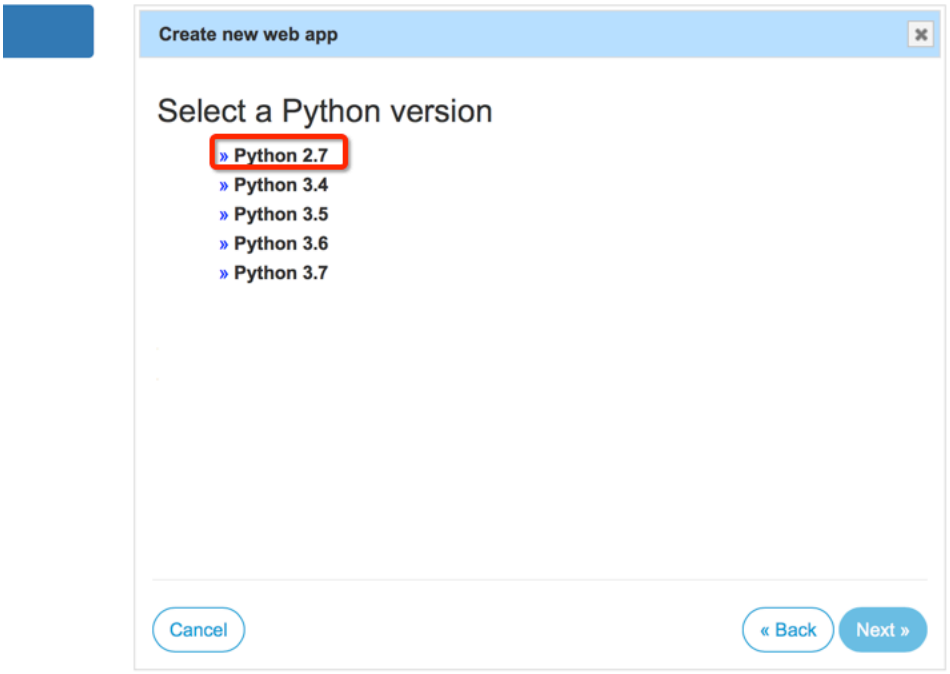
install mysql-client which is needed for accessing Mysql database on Pythonanywhere environment:

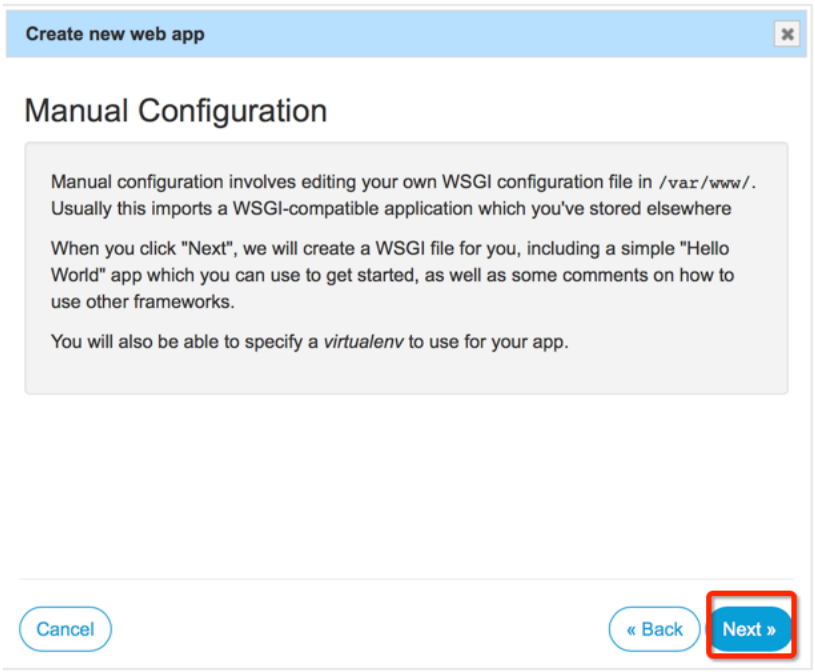
```
pip install mysqlclient
```

(5) add a web app

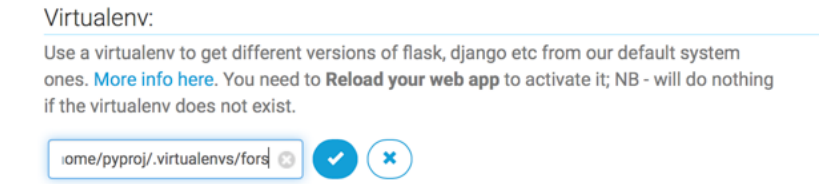



Add a new web app
Create new web app
Select a Python Web framework
...or select "Manual configuration" if you want detailed control.
» Django
» web2py
» Flask
» Bottle
» Manual configuration (including virtualenvs)
What other frameworks should we have here? Send us some feedback using the link at the top of the page!
Cancel
« Back
Next »


Create new web app
Select a Python version
» Python 2.7
» Python 3.4
» Python 3.5
» Python 3.6
» Python 3.7
Cancel
« Back
Next »

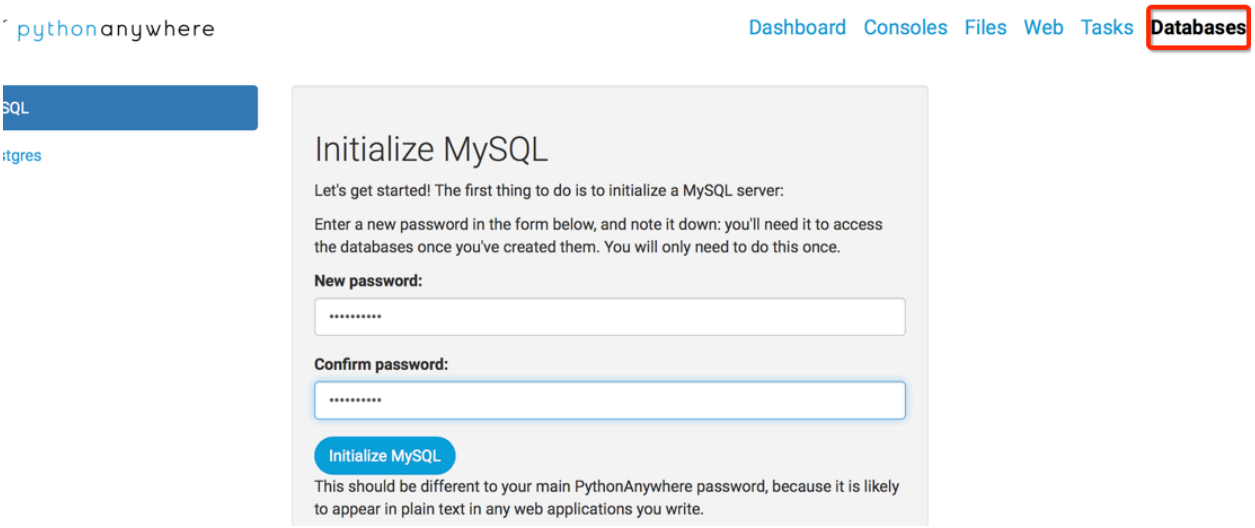


use the virtualenv created before:

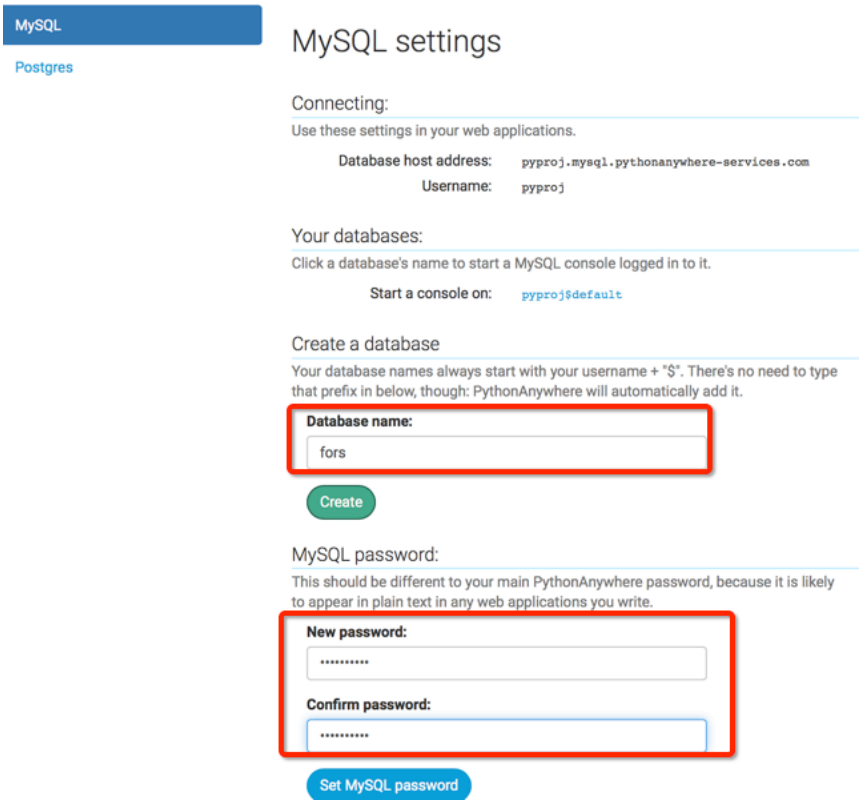


(6) Database Configuration

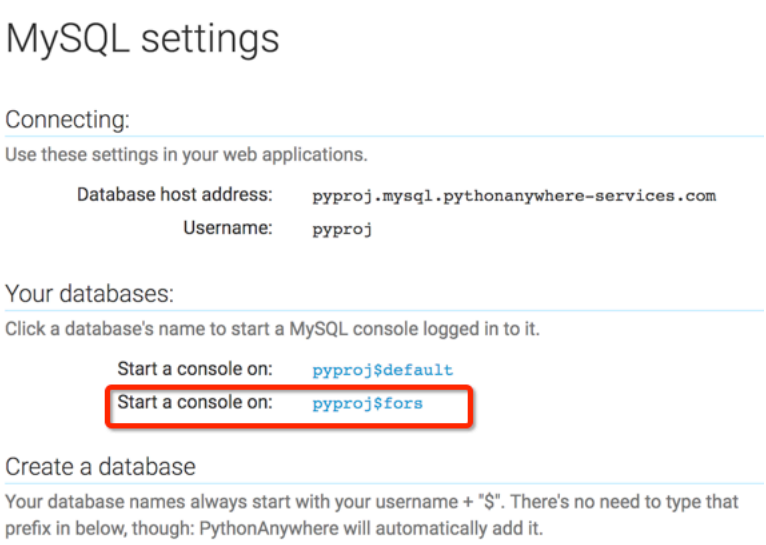
Next, we set up the MYSQL database. In the Databases tab of your PythonAnywhere dashboard, here we set a new password and then initialize a MySQL server:



then, create new database named fors. The database user is your username of PythonAnyWhere by default. If you want to change the passwod for the database later, you can also re-setup the password for your flask app in this page.



database created



(7) setup project env required for running under production env.
in bash console:

execute commands below:

```
export FLASK_CONFIG=production
export FLASK_APP=run.py
export SQLALCHEMY_DATABASE_URI='mysql://pyproj:your-password@http://pyproj.mysql.pythonanywhere-services.com /pyproj$fors'
```

The username and password can be found in the Database tab on your dashboard.
the format for SQLALCHEMY_DATABASE is
mysql://YourUsername:password@ YourUsername.mysql.pythonanywhere-services.com/ YourUsername $YourDatabaseName

(8) WSGI File

use WSGI file which pythonanywhere uses to serve the flask app. (replace the your-username, your-password, your-host-address and your-database-name to the corresponding values)

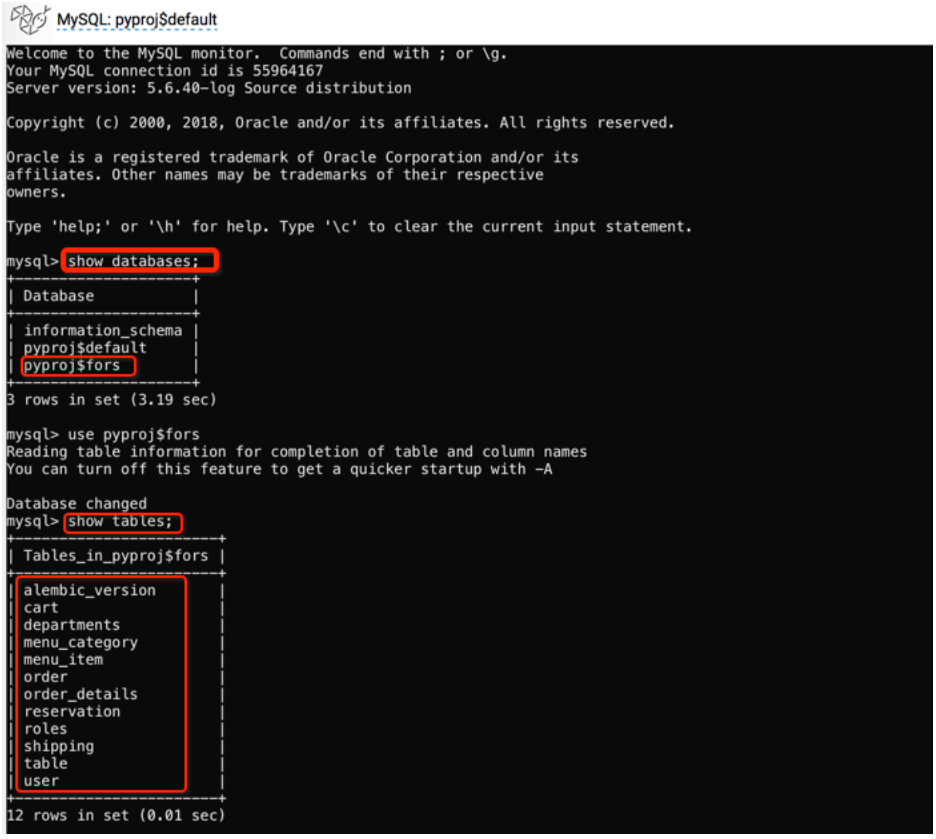
```
import os
import sys

path = '/home/pyproj/311proj'
if path not in sys.path:
    sys.path.append(path)

os.environ['FLASK_CONFIG'] = 'production'
os.environ['SECRET_KEY'] = 'abCv<XC889>$%1012'
os.environ['SQLALCHEMY_DATABASE_URI'] = mysql://your-username:your-password@your-host-address/your-database-name'

from run import app as application
```

(9) create tables using flask migrate:

execute these commands:

```
cd /home/pyproj/311proj
workon fors
flask db init
flask db migrate
flask db upgrade
```

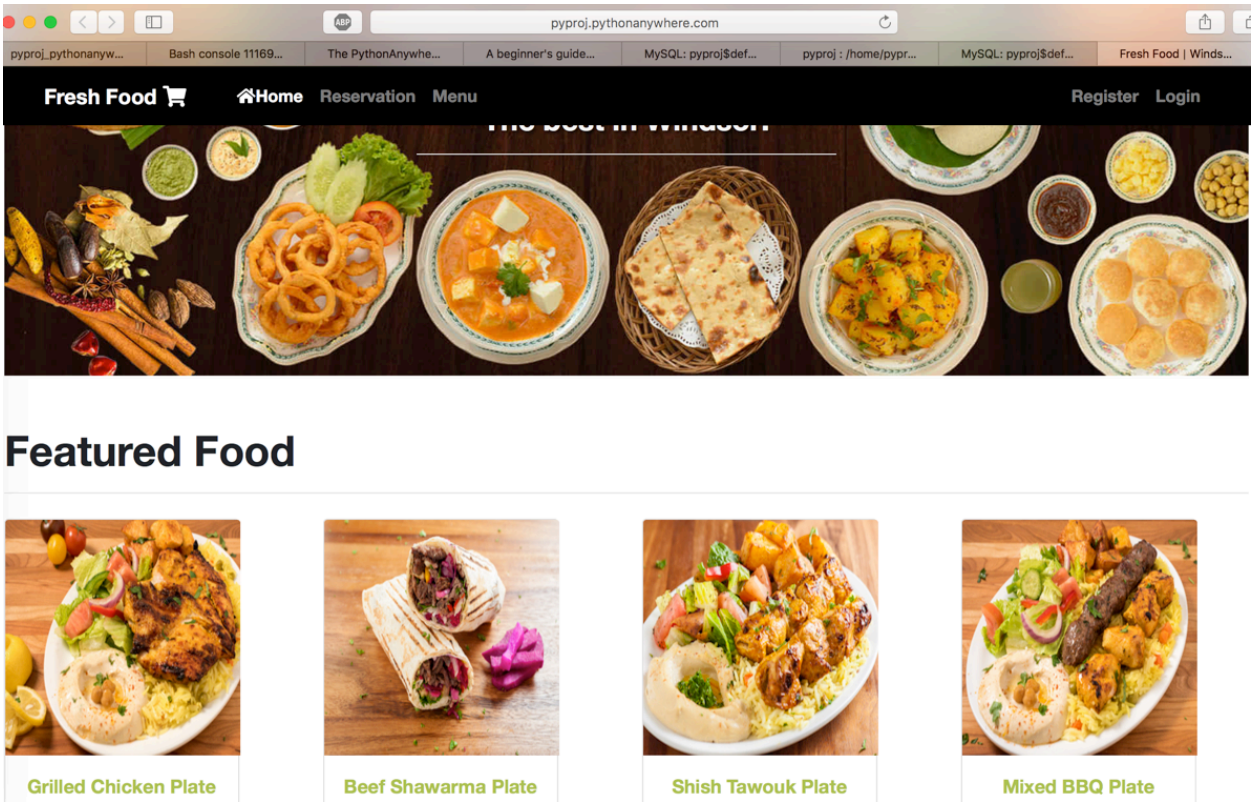
check if tables are created now



As you can see from screenshot above, tables of our project have been created successfully.

(10) In mysql console, we import data from the sql file which is exported from local mysql database in development environment.
mysql> source fors.sql

(11). access our app deployed on Pythonanwhere via URL:
http://pyproj.pythonanywhere.com (If you meet 500 error while access this URL, please refresh the web page and try again. This error does not occur in the development environment. We will figure out the reason for this.)



二. User Story Review

[5 Points]: List any changes in existing user stories, or any new user stories that you did or plan to the original user stories you submitted before.

user stories below from last submisson: **Group 9 Project Deliverable 2**

1. As a user, I want to clearly see the details of every food in the menu such as prices, calories, ingredients , so I can compare to each other, choose the best choice I wish to order. (card: 100, this is a very larger task because it includes more details in this implementation.)

5. As a user, I want to be able to edit my orders like change the items of food and units after I place an order. (card: 100, it's a very large task because if we change an order after it's been placed, then the work sequence of the staff must also be changed and conflicts might occur in this process.)

6. As a user, I want to be able to cancel my orders if I place an order mistakenly. (card: 5, this is a medium task because if the customer canceled the order before the restaurant accepts it then it would have no side effect.)

7. As a user, I want to to able to browse food by category or by rating. (card: 40, this is a large task because rating and sorting is a lot more extra work.)

8. As a user, I want to be able to select reserve a table or browser menu after login. (card: 10, this is a medium task since it may be nececssary to give an option for users to choose after login.)

original sprint planning from last submission:

6. Sprint Planning

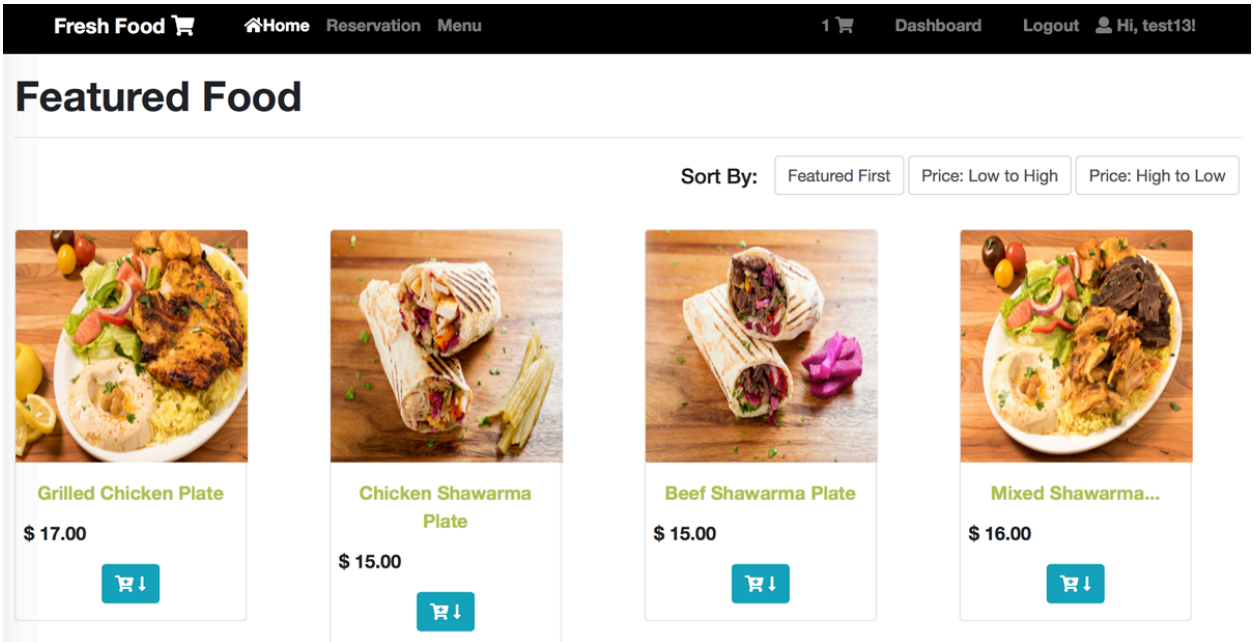
Sprint	User Stories No.	Time Estimation
1	1	5 weeks and more
	7	3 weeks and more
2	5	5 weeks and more
	6	3 day
	8	2 day
3	2	2 weeks
	3	1 weeks
	4	3 weeks

sprint 1	existing user stories	new user stories
	for user stories 1, 7, we implemented browsing menu and details of a food items, but browsing food by category or by rating is not implemented yet. We implemented user registration and login.	1. As a user, I want to register as a new customer in order to login in to place order or make a reservation online. 2. As a user, I want to login into the system use the account I registered in case that I can make a reservation by this system. implementation: We implemented login and registration function including login based different role like login with a admin user, a staff user and login with a customer user since the login and registration is the prerequisite of

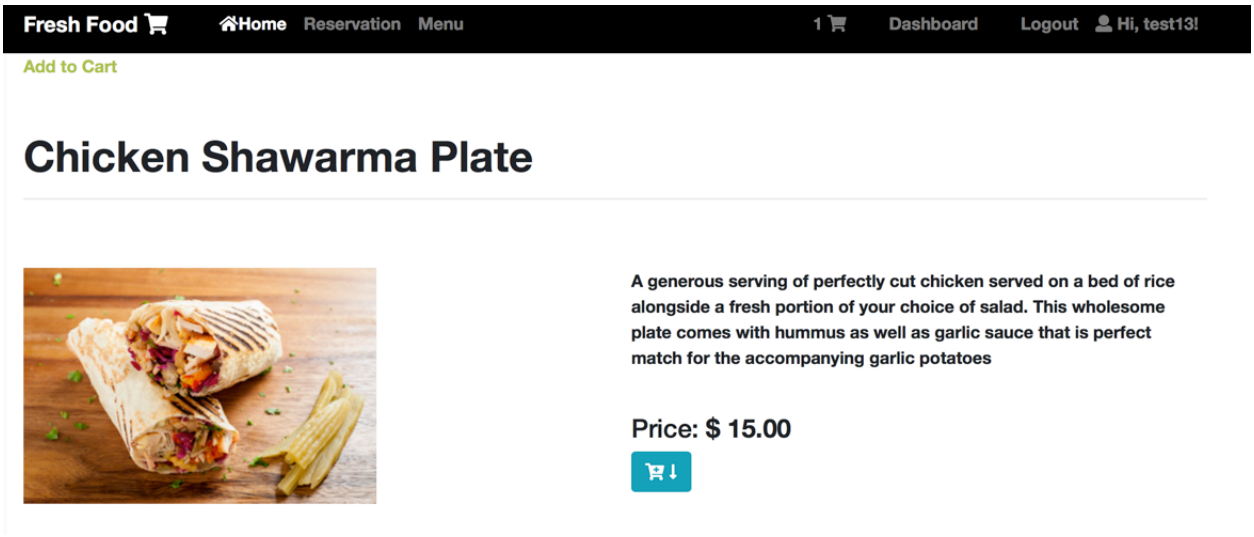
		all the user stories that need a user login to operate admin user can also manage deparment information, assign employee to a department and assign a role to a employee.
sprint 2	implemented user story #8, also include viewing the all the reservaton information including the reservation made before.	implemented a shopping cart in sprint 2, but it does not include the stage after clicking checkout button including filling the shipping address and view the order placed. These will be implemented in the following sprints.
sprint3 (planning)	As a user, I can place an order and then view the order status. As a user, I want to track the shipping status of an order. As a user, I can views the order list I have placed in the past. As a staff, I want to manage order in the system including view the details of an order, changing, canceling tracking the status of an order.	

You can see implementations from screenshot below:

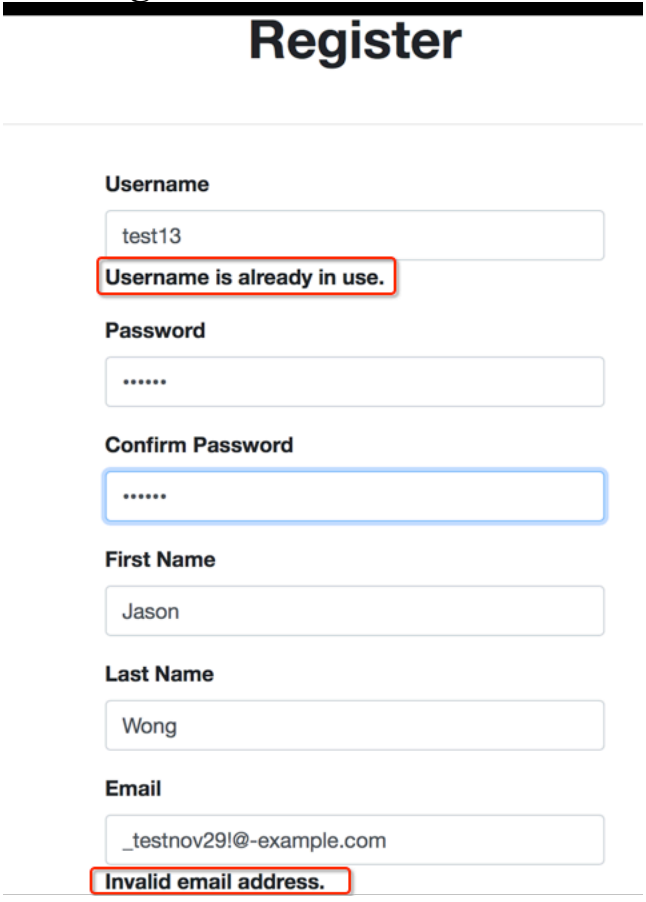
browse the food item list



views the details of a foo item:



user registration form with basic input validation:



user login:

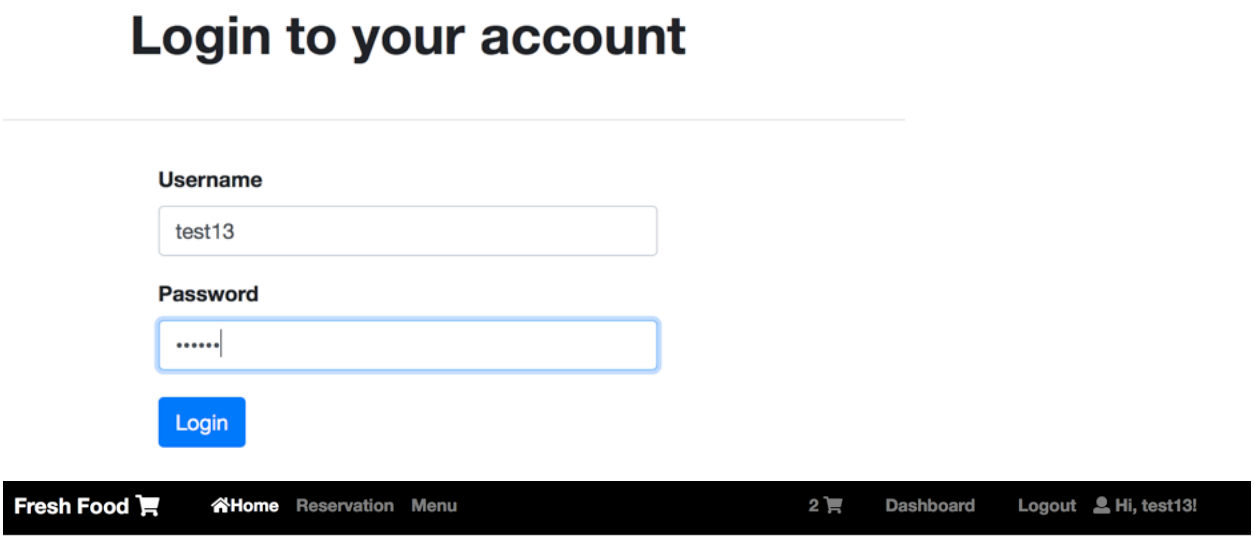
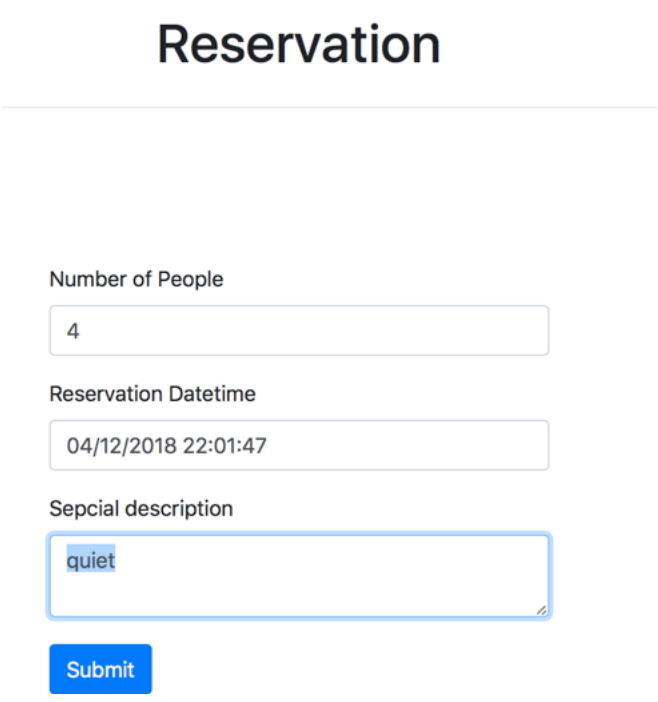
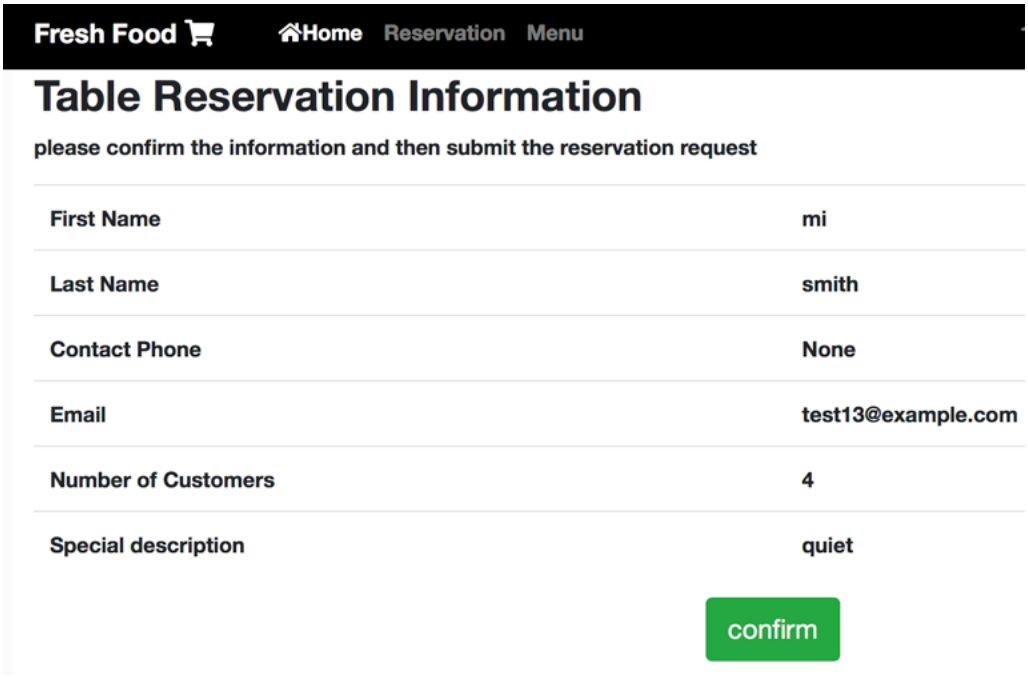
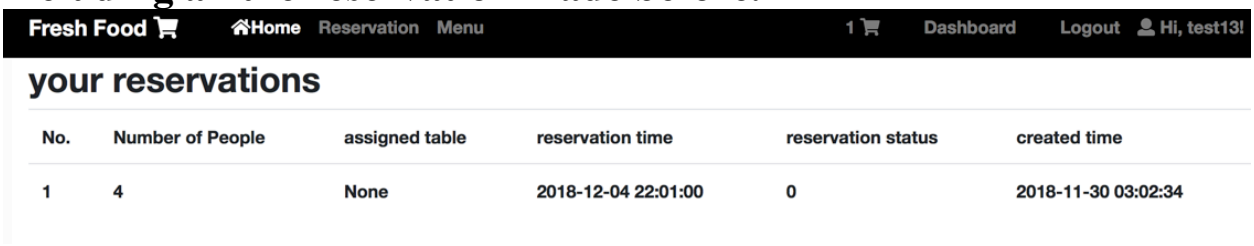


table reservation:

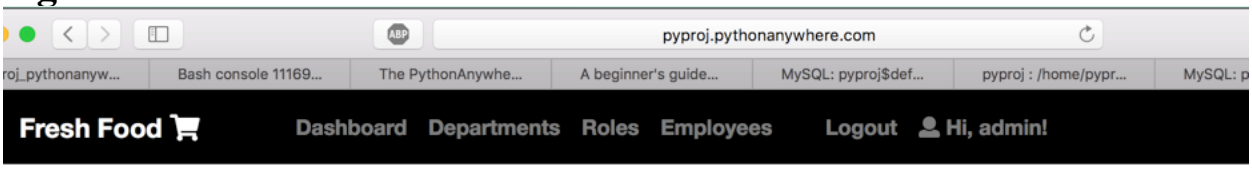




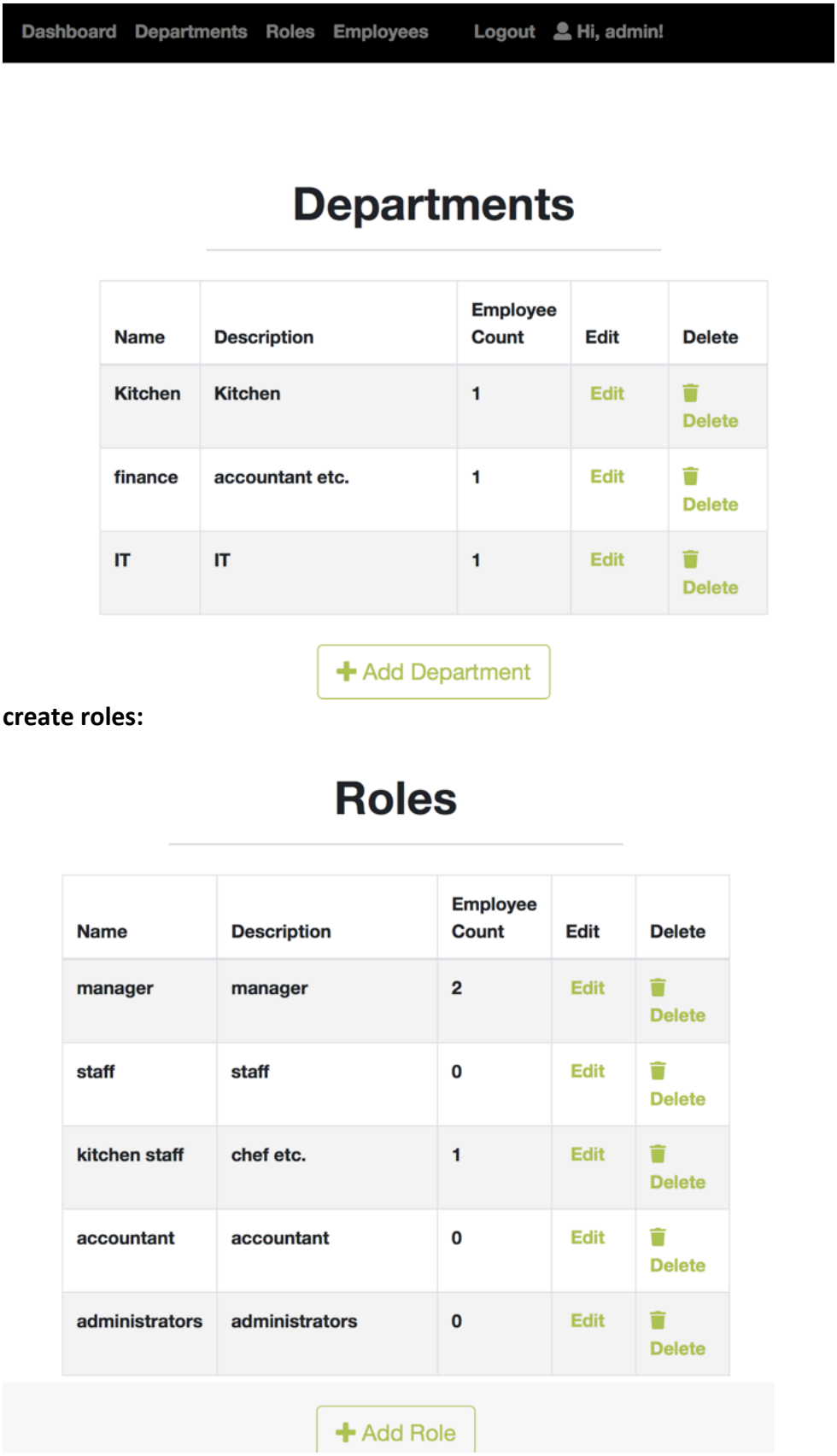
view the reservation information after making a reservation including all the reservation made before.



login as admin:

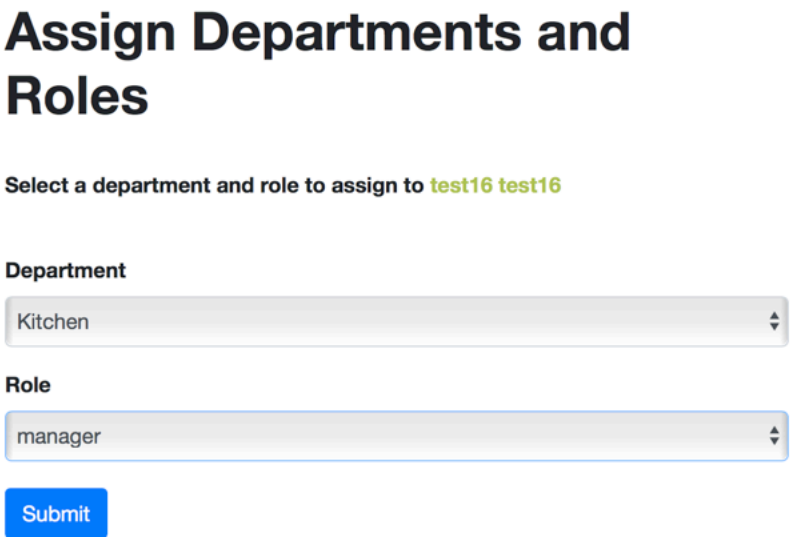


manage department:



create roles:

assign roles to a user and add a user to a departments



三. Design Review

[10 Points]: List any changes that you did, or you plan to do in the future to the original design you have submitted in part 2

changes that we did	changes that we plan to do in the future
(1).At first, we planned to use raw sql to perform queries since it can save time and improve efficiency, but we then found it very hard to well maintain the code and it also produce a lot of duplicated code. Therefore adopting a ORM framework called SQLAlchemy, which helps us better define the database tables and modeling the table to an class. It makes our software reusable and improves the maintainability.	(1) we will use github and manage our project and write wikis to record the process of problem solving and track the problem with issues. More importantly, we will use github to integrate with automatlly testing and deployment. By doing these, our efficiency of development and testing will improve significantly
(2). In the past design stage, we did not consider the shopping cart, which is very important for this kind of system. In the implementation, the shopping cart and its related classes have been implemented.	(2) we will introduce an Publish–subscribe messaging pattern and integrate it into our system. By doing this, the customers users, staff including managers, chef, host and etc. will receive message instanly once the senders send a message or triger an event. It

	also helps achieve a high cohesive and loosely coupled system.
(3) In the sprint 1, we implemetanted storing user password using bcrypt password hashing, which is much more better than md5+salt. Therefore ensuring a better security of password storing.	
(4) in this implementation, we organized our program by dividing it into different modules including admin, auth, cart, home, menu, order, reservation and shipping, which make the code more readable and more maintainable.	
(5) In terms program config files, we defined three different config files corresponding different environment: development, production, testing. This will make it easy to deploy and run our project in different scenarios.	

四. Implementation Review

[5 Points]: List any changes in your technology stack

(1). As you can see from Execution Instructions, we introduce an Python ORM framework as an abstract layer used in python to interact with Mysql database instead of executing raw sql.
(2). we use javascipt in the client (browser) side to implement better user experience and provide some feature like ajax support and datetime picker.
(3). We also use some javascript library like jquery (this is required to support other component like datetimepicker, bootstrap and etc.), bootstrap datetimepicker (this is used in datetime selection in table reservation form)

五. Team Velocity

[10 Points]: Report your team velocity and compare it to your original estimation in part 2. Provide a new estimate for the user story if required.

original sprint plan:

6. Sprint Planning

Sprint	User Stories No.	Time Estimation
1	1	5 weeks and more
	7	3 weeks and more
2	5	5 weeks and more
	6	3 day
	8	2 day
3	2	2 weeks
	3	1 weeks
	4	3 weeks

user stories below from last submisson: **Group 9 Project Deliverable 2**

1. As a user, I want to clearly see the details of every food in the menu such as prices, calories, ingredients , so I can compare to each other, choose the best choice I wish to order. (card: 100, this is a very larger task because it includes more details in this implementation.)

2. As a **staff**, I want to be able to edit customers' order, prevent the mistakes are made by us. For example, a customer reserve a table in the restaurant, but the reservations are full then we need to cancel. (card: 20, this task is large because to achieve this we need to implement an authentication and permission service first.)

3. As a **staff**, I want to see the orders I need to prepare so I can schedule my work sequence more efficiently. (card: 15, this is a medium task because scheduling staff's work sequence is a little challenging.)

4. As a staff, I want to edit the menu, so customers can see what is new.(card: 40, same reason as (II).)

5. As a user, I want to be able to edit my orders like change the items of food and units after I place an order. (card: 100, it's a very large task because if we change an order after it's been placed, then the work sequence of the staff must also be changed and conflicts might occur in this process.)

6. As a user, I want to be able to cancel my orders if I place an order mistakenly. (card: 5, this is a medium task because if the customer canceled the order before the restaurant accepts it then it would have no side effect.)

7. As a user, I want to to able to browse food by category or by rating. (card: 40, this is a large task because rating and sorting is a lot more extra work.)

8. As a user, I want to be able to select reserve a table or browser menu after login. (card: 10, this is a medium task since it may be nececssary to give an option for users to choose after login.)

New user stories:
N1. As a user, I want to register as a new customer in order to login in to place order or make a reservation online.
N2. As a user, I want to login into the system by using the account I registered in case that I can make a reservation by this system.
N3. As a user, I want to add a food item into my shopping cart so that I may place an order later.

Sprint	User Stories No.(Planned)	New User Stories No.	User Stories No.	Time Estimation(original)	Time Spent
1	1		1 (100%)	5 weeks and more	2 weeks
	7		7 (70%)	3 weeks and more	1 week
		N1 (100%)		2 weeks and more	1 week
		N2 (100%)		1 weeks and more	3 day
2	5		5 (0%)		change to updated sprint 3
	6		6 (0%)		change to updated sprint 3
	8		8 (100%)	2 day	3 day
		N3 (100%)		3 day	2 day

updated sprint 3 (this will be implemented in the future)

Sprint	User Stories No.	Time Estimation(original)	Time Estimation (new)	Description
3	2	2 weeks	1 week	
	3	1 weeks	3 days	
	4	3 weeks	1-1.5 weeks	
	5		3 days	move from sprint 2 (It will be

				implemented in shopping cart rather than in orders
	6		1 day	move from sprint 2

.