# HW2 report

Name: Yanan Zhang, Ziyi Mu

UNI: yz3054 and

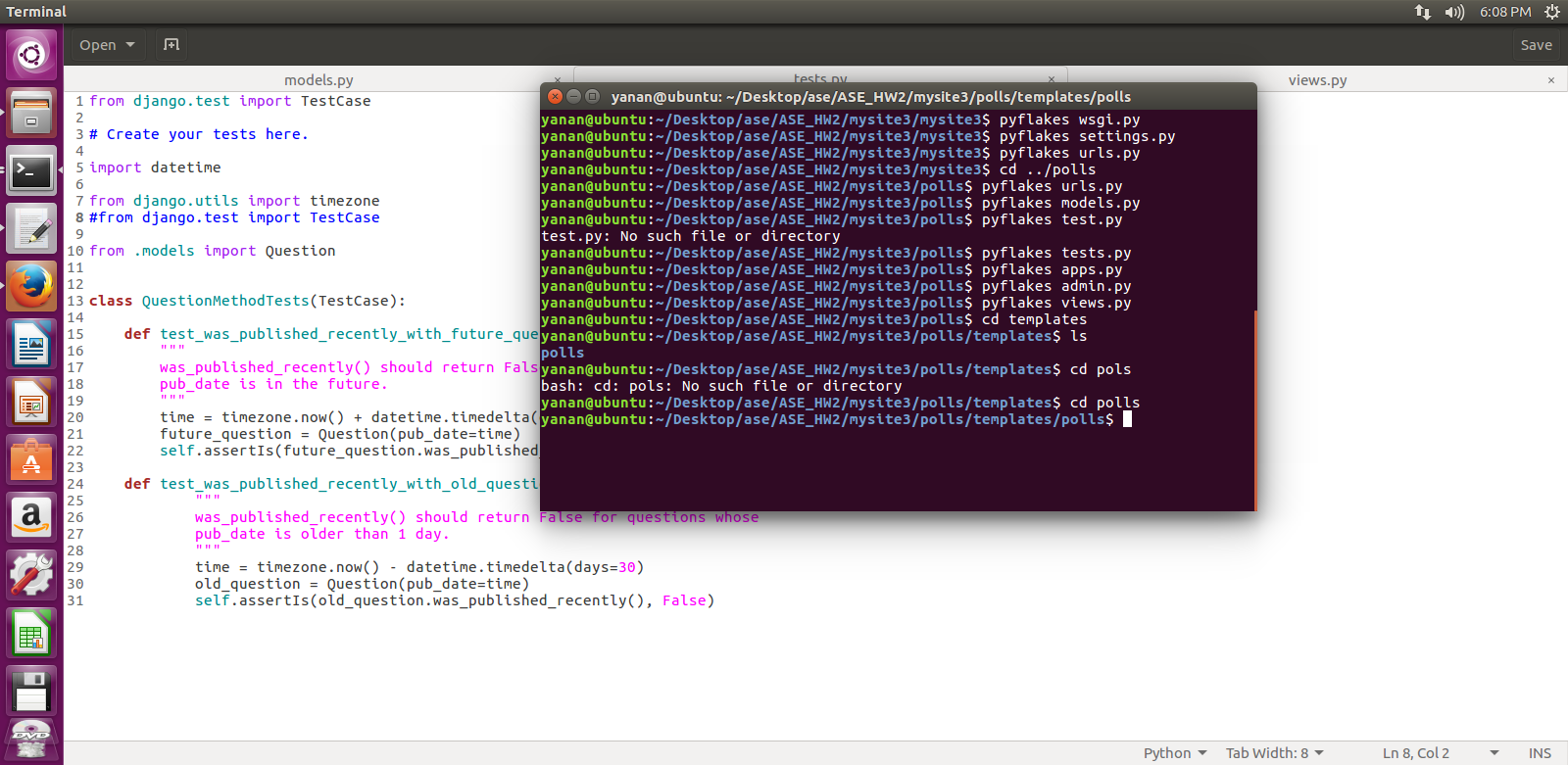
1. **System description**

In this assignment, we built a toy voting system using Django web development tool with python language. The function of the app is to collect the votes and post the questions and collect the votes. The data for questions, votes, and choices are stored in a SQLite database. We also integrated the HTML templates to the web development to edit the layout of the webpages and improve the views. The project name is newsite3 and there is a polls app built in this project. We developed this toy system using Django in python3 environment under Ubuntu 16.4 platform (supported by vmware virtual machine).

To use this, follow the steps:

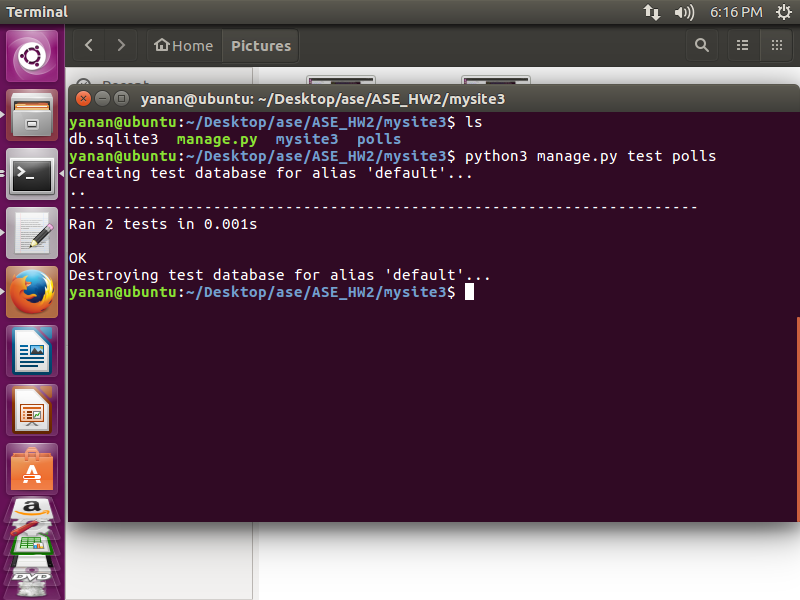
1. locate the path to ASE\_HW2/mysite3/
2. Execute the command: $python3 manage.py runserver
3. The users also could using shell to run/edit. Execute the command: $ python3 manage.py shell
4. **Code checking**

After we build the code, we use the code smell detector of pyflakes detector to check the code. After several round revisions, now we have the code PASSED by the pyflakes detector.



1. **simple tests**

For the tasks, we build 2 tests for the web development and both of them are OK. The two tests are related to the question posting date. The two tests results screens are presented below:



1. **Experiences**

At first, we want to install python2 with Django and found it needs additional work when installing Django on python2.In addition, it usually reporting errors. So it is easier to use python3 environment along with Django and we switched to python3 ultimately. But there are some syntax differences between python2 and python3, people used python2 a lot need to remind the differences frequently.

It needs to build the url linking for the project as well as the polls. At first, we only built the urls linking in the project and forget to link the app urls to the project. The results show that we could not implement the app function at all! After debugging, we fixed the problems and run the application successfully.

For Django, migration is very important, we run into problems such as missing migration which caused the Django could not link the database models and web views correctly.

Due to the time limitation, the layout is very simple. For next step, we need to improve the look of the app and also need to add more testing cases.

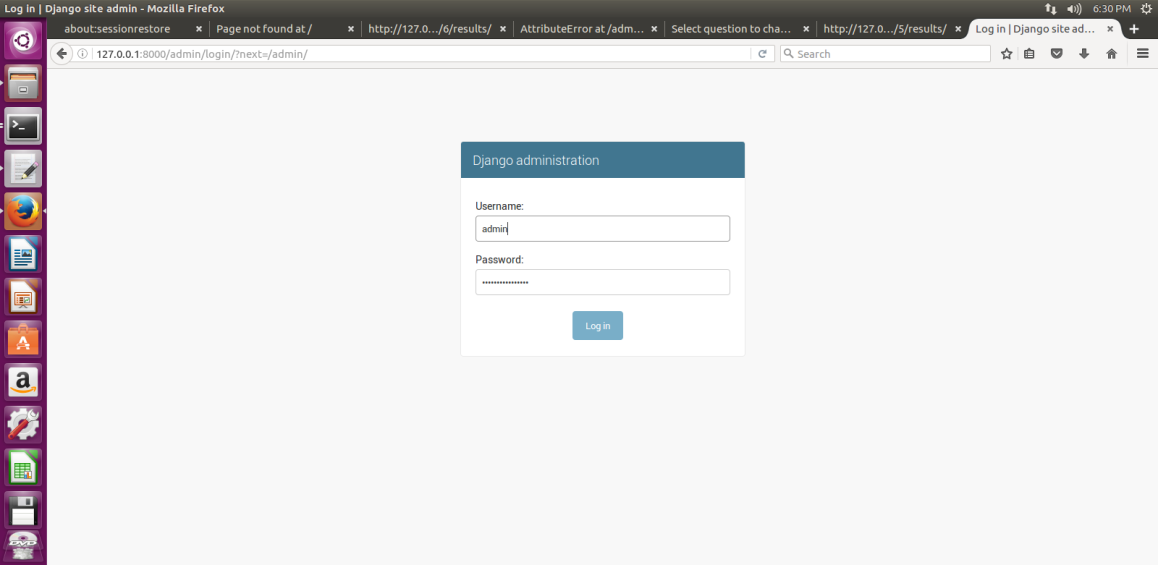
1. **Appendix**

There are 4 views for the application which are admin, index, detail and results. The admin is used for manage the voting system where the super user could login to edit, add, or delete the questions with their voting choices.

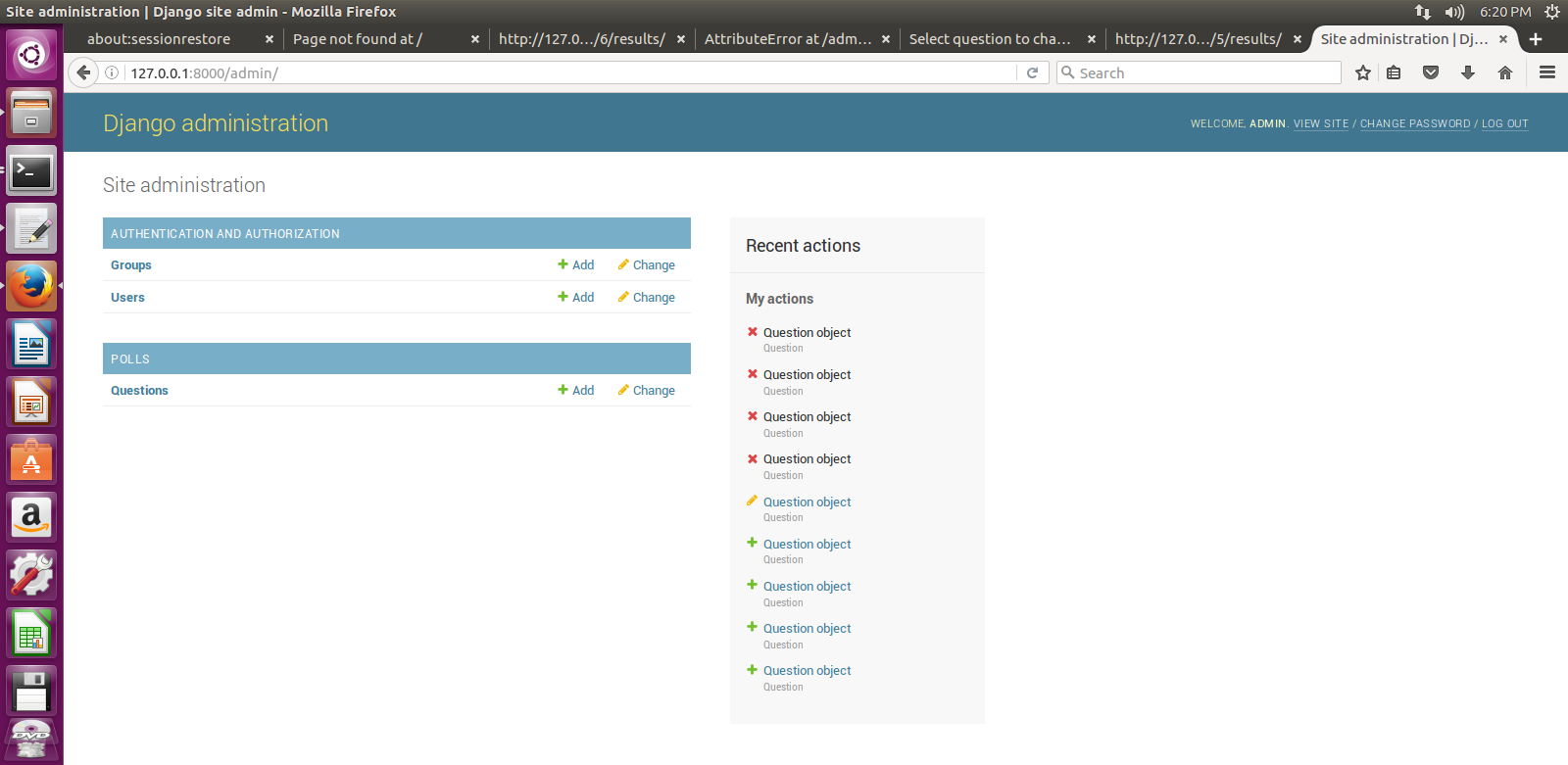
**Admin view**

The admin view consisted of several linked pages. The main url is: localhttp/amin

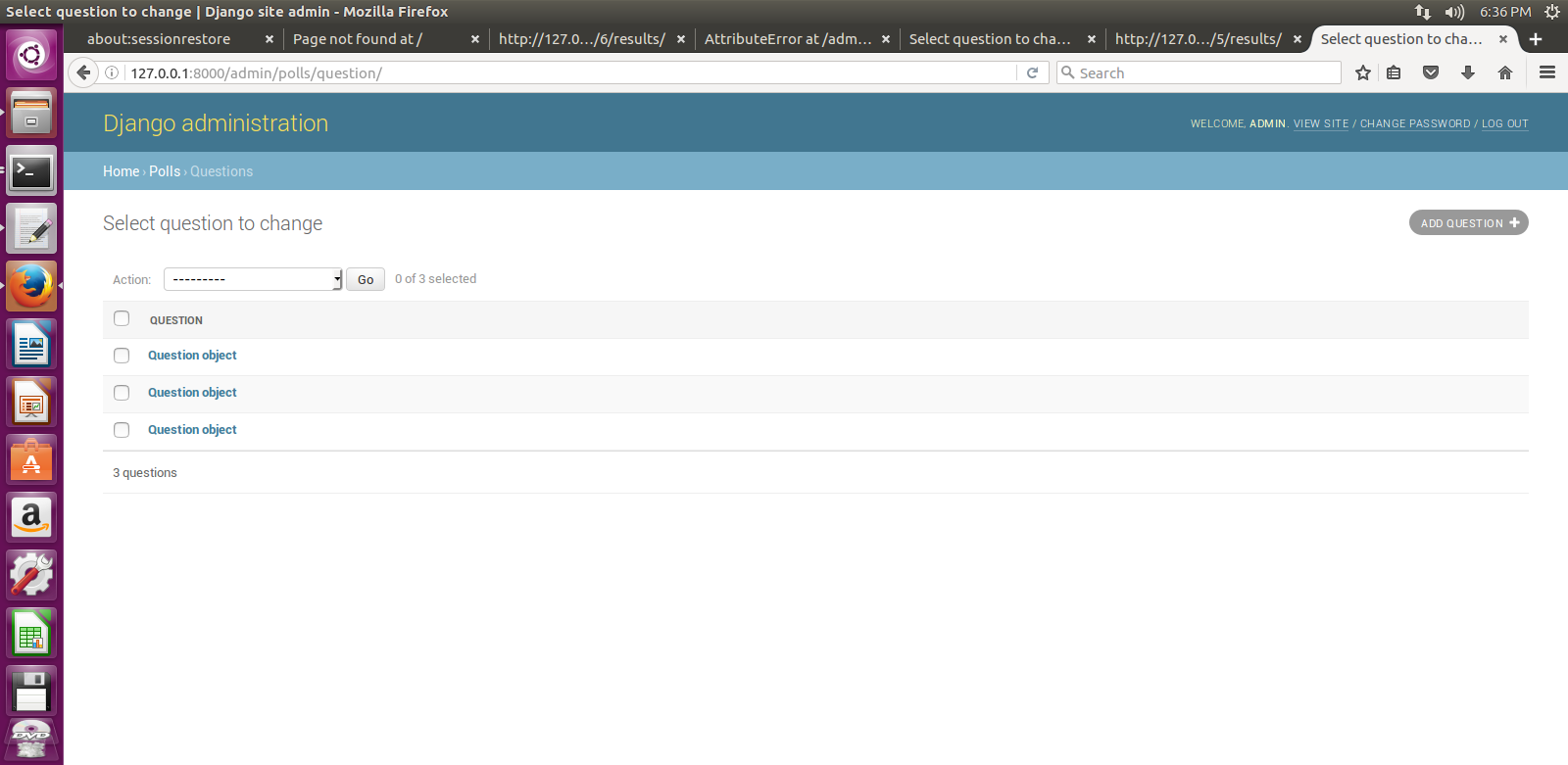
1. When the super user login, he/she need to type the user name and password. The Username is admin and Password is zyntop0138682812.



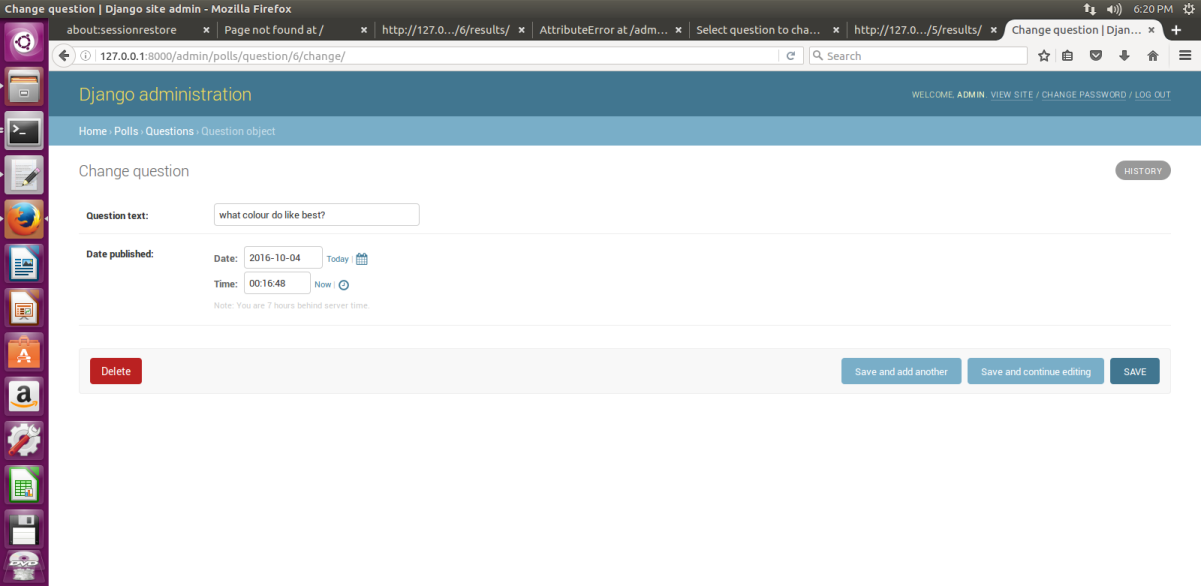
1. After the user login, then he/she could edit, add or delete the questions.



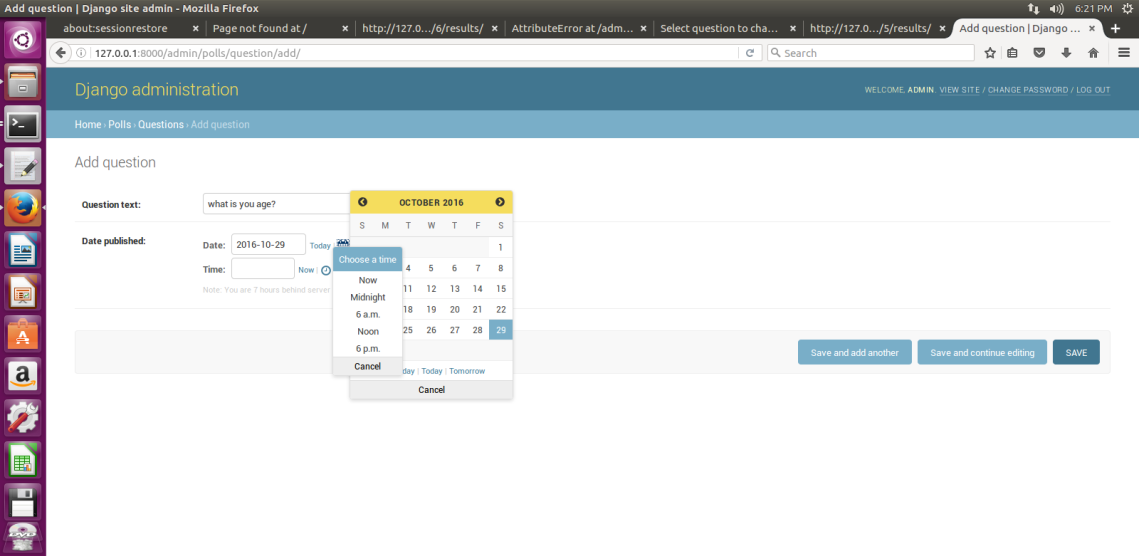
1. There is a list of questions, the user could click on them and edit.



1. This is one of the sample questions in this system.



1. The user could also add new questions.

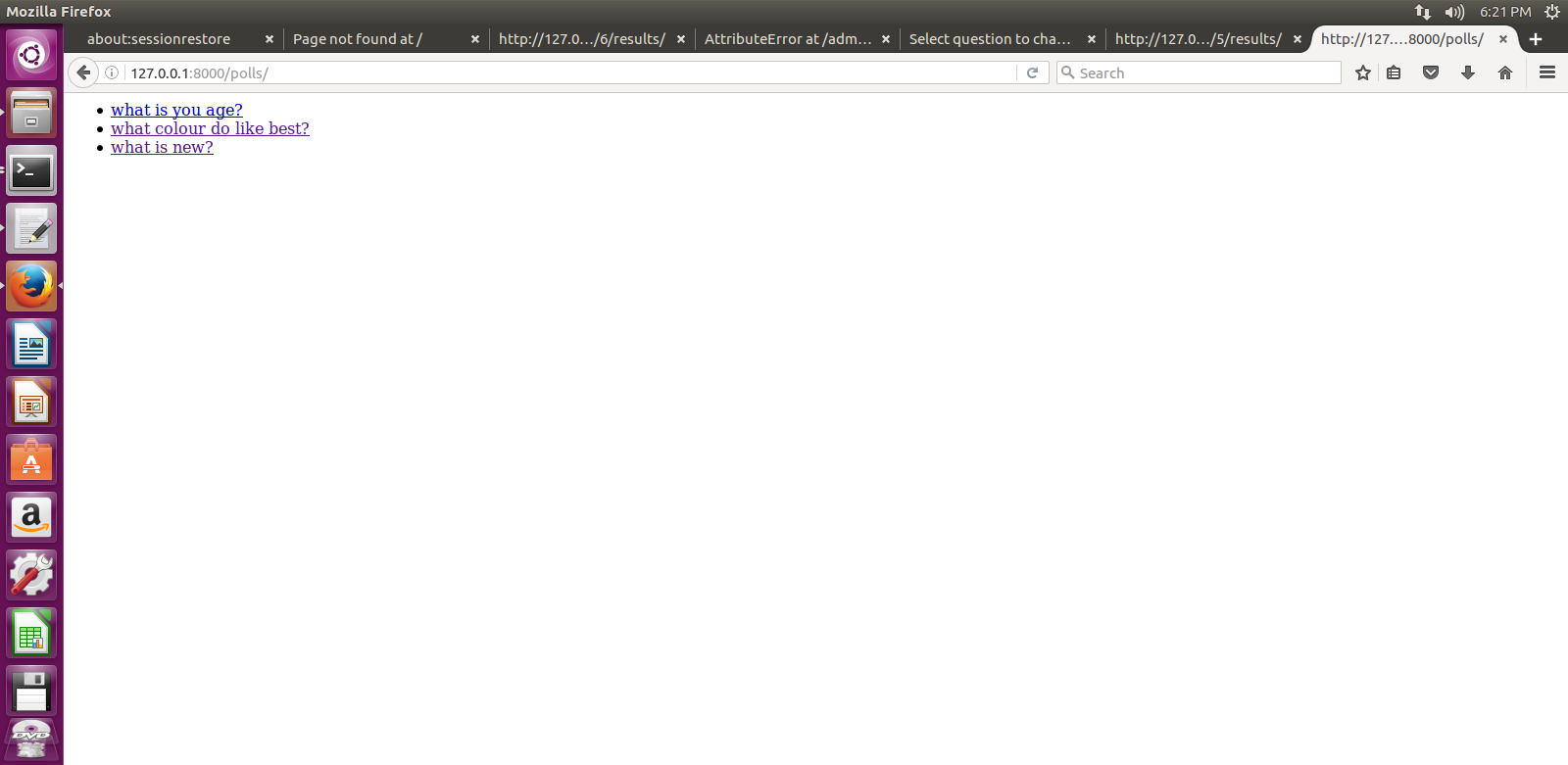


Polls Views

There are index views, details view and results view associated with polls views. The url is localhttp/polls, which is different from the admin url.

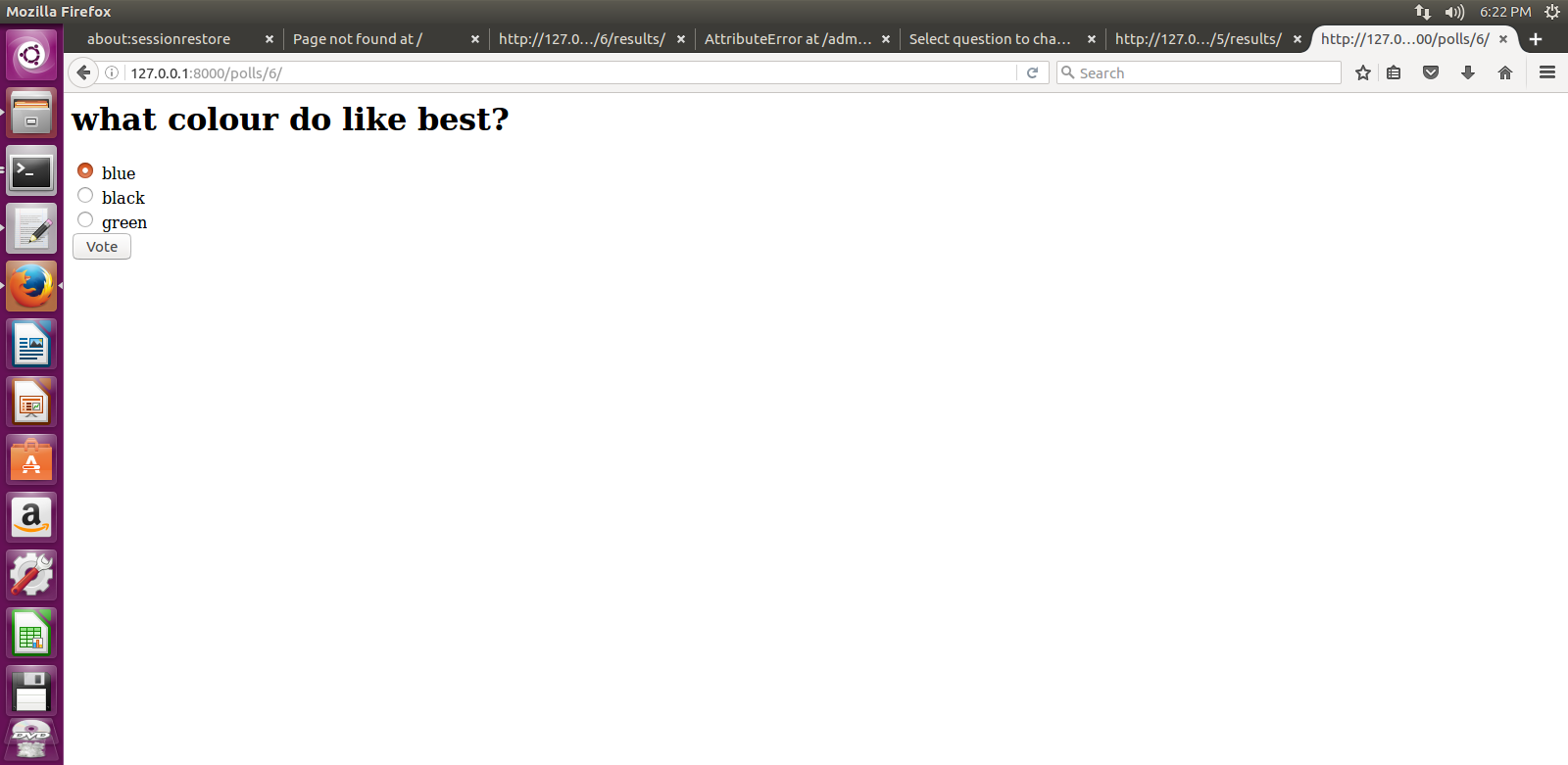
1. Index view.

In this index view, the web shows the total questions. Click on each link will direct the clients to the details view.



1. details view

When the clients click on one of the questions, the web will be directed to the detais view for the question. Here is the one example. It shows the question and the voting choices. The clients could vote by click on the radio buttons. The voting data will be saved to the database automatically.



1. results view

After the clients voted, it will direct to the web page for the voting results.

