Team Write Up

Title: ShootA
Subtitle: InfoTrack
Semester: Fall 2017

Overview: Our application is used to allow student sharing information with each other. Our application is innovative because there is no such website allowing users to share their information at campus. “油麦地”maybe a good resource for Chinese but not American students, and their section pages are not user-friendly. If we can have more time, we will elaborate our website for a more friendly-use UI, and implement functionality such as “like the post” and give all the post a ranking list.

Team Members: QIQIN ZHAO, FENFANG DONG, ALAN CAI, YIFU LIU, RUIFENG WANG

Github Repository: https://github.com/RoyceDavison/Finalsubmission

User Interface: I will include screenshots of each UI views in our github final submission repo. Homepage can redirect login user into other sections, and prevent anonymous users access any other section in our webpage.
ClubInfo: Information about club information such as Chinese club have a meeting on Friday or Finance Club have a party tonight.
CourseInfo: Users can find easy class recommend by the previous students. For example, Chinese 121 is easy A.
Freeride: Users can find free carpool in this section. For example, a user may want a free ride with other students go to logan airport.
RenInfo: Users can find renting apartment information in this section.
TutorInfo:Users can find tutor information in this section. Such as Chemistry121,131 private tutor $20/hour.
View_profile and edit_profile: Allow users to view his/her profile and make change on it.
Change Password and reset password: if a user forget his/her password, he/she can reset password by sending email. Change password allows users to change his/her password.
Make a new post: Allow users to make a new post in the corresponding section.
Read more post: Allow users to read all posts.
Search: Allow users to search specific post information.
Add comment: Allow users to add comment to specific post. If you like the tutor, you can add comment in the privatetutor section to support the tutor.

Data Model:
I include the diagram in github_ FinalSubmission. And I will briefly describe the model here. Our final data models are uesrprofile, post and comment.

User Profile: profile links to user(django built-in). This model includes location,website,description,phone,userid,birth date, image, studentID, grade, and major. This model has one to one relationship to User(django built-in model).So when users register their account, their profiles will be saved automatically into this database.

Comments: This model includes context, image, video(URL), time, user(as foreignkey to identity the comment author), and has one to many relationship to post, since a post can have many comments. So when a user add comment to a post, it get automatically saved into the database. We have context, image, video,time(created time) for this model.

Posts: This model includes title,context,time,video(URL),posttype(used to classify the post),user(as foreignkey to identity the post author), and it has one to many relationship to User, since a user can make many posts. So when a user write a post, it get saved into the database.

There are maybe some other models in our admin, but we decide to leave those models to future works, since we don`t have enough time to implement all of those.

URL Routes/Mappings:
We group our URLs according to their functionality.
Homepage allow use to view home page, everyone can view this page include anonymous users.
Test is used by our team member to test functionality of the website.
Five sections: clubinfo/courseinfo/freeride/privatetutor/rentinfo are used by users to view specific information as the section name suggested, the big giant box in each sections include the most five recent posts in all five sections.
We also include two url for profile: view_profile and edit_profile. Like the name suggests, these two URLS will redirect the users to profile page and allow the user edits its profile.
The rest two big sections are related to password and post & comment.
Change_password & reset_password & password_reset_done & password_reset_complete & password_reset_confirm are all related to set password, like the name suggested the URLs will redirect the user to related template. The reasons for the last three URLs because we want to overwrite the django build-in template for resetting password, so we can customize our webpage more friendly used.
The rests are related to post, add_post allows use to make a post on our website. Post_list directs user to view all posts already made in our website. Add_post allows users to add a new post. Post_detail allows users to view all context of a single post he/she selected. Post_search allows user to search for specific post. Add_comment allow user to add comment to a selected post. Post_edit allows user to edit his/her post, but we are undecided on whether to implement it.

Authentication/Authorization: Our website only allow login users to access. Suppose you are anonymous user, you click any five sections from homepage you will be redirect to “Page Not Found”. After you log in, you can view all information on all sections.

Team Choice: Our team choice is when a user forgets password s/he can ask reset email by sending email. But when you change email, you should also change manually on ShootA views.py. We currently use qzhao@umass.edu as recipient.

Conclusion: Our team are really good at communicating with each other, and all members showed up in every meeting. We are proud of two html we made, one is the homepage and other is the sections based generic base page. The implementation and design process takes so long for us, because some of us are not doing their jobs and one member is not knowing python so it is a difficult for our team implementing many functionality. We would like to know the model and form design before starting the project since we had a huge problem in linking model at the beginning. The most difficult problem our team encountered is two people are not doing their supposed server-side implementing jobs, and one person not knows python language and only has minimum understanding of the programming. Only two people implement the server-side who don`t have any previous experience with django.