

Overview

Introduction

During the course so far we have developed a number of web servers using Django. This has covered single page applications, database schemas, appropriate model design, forms and templates, RESTful webservices, Web Sockets, Authentication and many other topics. For this assignment you are tasked with developing a Social Network App using all knowledge you have gained on the course so far. This assignment is worth 50% of the total mark for this module

Task

To implement your own social network web application. The minimum application requirements are:

- Your application should allow users to create new password secured accounts.
- You should collect and store an appropriate amount of information about each user (e.g. user name, use picture).
- Each user should have a “home” page that shows their user information and any other interesting data such as images, picture galleries or other media files. And displays user status updates. These home pages should be discoverable and visible to other users
- Users should be able to post status updates to their “home” page.
- Users should be able to search for other users and “add” them to their network of contacts or friends.
- Users should also be able to view a list of their friends.
- Additionally users should be able to use a portion of the application that makes use of real time communication by using web sockets. Some examples may be realtime chat between friends, a basic computer game or media/audio streaming. You are free to use web sockets as you wish but you must include 1 web sockets app in your application
- An appropriate REST interface for User data should be provided.
- The application should include functionality that makes appropriate use of Web Sockets

You are free to design the application layout as you wish. Each functional module (friend list, chat, search results, etc...) may be a separate page or you may choose to make the application a Single Page Application.

Deliverables

1. A django application that implements the a social network web application and fulfils the functional requirements. The Application should include some users for demonstration purposes
2. A report (4000-6000 words) describing the application and the reasoning for its design and functionality. The report should explain how your application meets the requirements. Explain the logic of your approach, why is your code arranged as it is? Explain the design of your application, covering all design and implementation decisions you have made. The end of the report should critically evaluate your application. Discuss your design and implementation, did it work well? what parts of the application could be better? What would you change if you attempted the project again? This report should also include a brief section on how to run the unit tests.

Requirements

We will assess your work based on the following requirements and criteria:

- R1: The application contains the functionality requires
 - a) Users can create accounts
 - b) Users can log in and log out
 - c) Users can search for other users
 - d) Users can add other users as friends
 - e) Users can chat in realtime with friends
 - f) Users can add status updates to their home page
 - g) Users can add media (such as images to their account and these are accessible via their home page
 - h) correct use of models and migrations
 - i) correct use of form, validators and serialisation
 - j) correct use of django-rest-framework
 - k) correct use of URL routing
 - l) appropriate use of unit testing

m) An appropriate method for storing and displaying media files is given

- R2: Implements an appropriate database model to model accounts, the stored data and the relationships between accounts
- R3: Implementation of appropriate code for a REST interface that allows users to access their data
- R4: Implementation of appropriate tests for the server side code

Code style and technique

Your code should be written according to the following style and technique guidelines:

C1: Code is clearly organised into appropriate files (i.e. view code is placed in an appropriate view.py or api.py file, models are placed in an appropriate models.py file)

C2: Appropriate comments are included to ensure the code is clear and readable

C3: Code is laid out clearly with consistent indenting, ideally following python pep8 standard

C4: Code is organised into appropriate functions with clear, limited purpose

C5: Functions, classes and variables have meaningful names, with a consistent naming style

C6: Appropriate tests to cover the API functionality are provided

Submission

You should write a brief report and submit your source code. The submission should contain the following items and information:

D1: Django code in standard ZIP format

D2: A report in PDF format. Including how to unpackage and run your application and how to run the tests for your application.

Review Criteria

We will mark your work according to the set of criteria shown below, which consider the requirements, your programming technique and style and the documentation you have provided:

The following marking rubric will be used to assess the application and the report:

