BSc Computer Science, Advanced
CM3035 Advanced Web Development

Endterm Coursework: A Social Network

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.

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 chat in real time to the contacts of their choice.
- 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 social network web application and fulfils the functional requirements
2. A brief report 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? This report should also include 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 accesible 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 and 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 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.

Marking 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:

Category	**Criteria**	**Not addressed**	**Attempted but did not meet requirements**	**Met described requirements**	**Met requirements and went significantly beyond them**
Requirements	R1				
Requirements	R2				
Requirements	R3				
Requirements	R4				
Code style and technique	C1				
Code style and technique	C2				
Code style and technique	C3				
Code style and technique	C4				
Code style and technique	C5				
Code style and technique	C6				
Submission	D1				
Submission	D2				