Blog Lite

Problem definition

Modern Application Development - I

Last Updated: 22-09-15

Frameworks to be used

- Flask for application code
- Jinja2 templates + Bootstrap for HTML generation and styling
- SQLite for data storage
- All demos should be possible on a standalone platform like replit.com and should not require setting up new servers for database and frontend management

Blog Lite

- It is a multi-user app
- Used for uploading blogs with images
- User can post multiple times
- Each post will have
  - ID
  - Title
  - Caption/Description
  - ImageURL
  - Timestamp
- A user can follow other users using the app
- Each user will have
  - username
  - Password
  - No of followers
  - No of posts
- Every user will have its own feed
- System will automatically show the blogs from the users you follow in a particular sequence
- The recommended order of blogs in a user's feed is based on the timestamp of blogs

Terminology

- Social Platform
- Profile - Basic stats, List of blogs
- Feed - Lists of blogs uploaded by other users you follow
- Archive (optional) - Blogs can also be made private / hidden from others

Similar Products in the Market:

1. Instagram
   - Web, IOS and Android
2. Facebook
   - Open Source
   - Web, IOS and Android
3. Twitter
   - Open Source
   - Web, IOS and Android

- These are meant for exploring the idea and inspiration
- Don’t copy, get inspired

Example Wireframe

- Click this link to check the wireframes
- It is just given to gain a basic understanding, and not meant to be followed exactly

Core Functionality

- This will be graded
- Base requirements:
  - User signup and login
  - User profile view with basic stats
  - Blog Post Management
  - Search and Follow / Unfollow Others
  - User’s Feed

Core - User Signup and Login

- Form for username and password
- You can either use a proper login framework, or just use a simple HTML form with username and password - we are not concerned with how secure the login or the app is
- Suitable model for user

Core - User’s Profile

- Basic profile view for a user
- Ability to view the number of blogs created
- Ability to view the number of followers and people you follow
- Ability to view the list of posts created

Core - Blog management

- Create a new blog
  - Storage should handle multiple languages - usually UTF-8 encoding is sufficient for this
  - Content should handle the safe HTML tags
- Edit a blog
  - Change title/caption or image
- Remove a blog
  - With a confirmation from the user

Core - Search and Follow / Unfollow Others

- Ability to search other users
- Ability to follow others
- Ability to unfollow others

User’s Feed

- Show the blogs/posts created by other users
- Navigate to the user’s profile on clicking the username on the blog or post

Recommended (graded)

- APIs for interaction with users and blogs
  - CRUD on users
  - CRUD on blogs
  - Additional APIs for getting the blogs/posts to show in feed
- Validation
  - All form inputs fields - text, numbers, dates etc. with suitable messages
  - Backend validation before storing / selecting from database
- Engagement on Blogs/Posts
  - Ability to like or add comments on a blog
  - Analyse the engagement of blogs/posts

Optional

- Styling and Aesthetics
- Proper login system
- Export blogs/posts engagement (number of likes/comments on each blog/post of a user)

Evaluation

- Report (not more than 2 pages) describing models and overall system design
  - Include as PDF inside submission folder
- All code to be submitted on portal
- A brief (2-3 minute) video explaining how you approached the problem, what you have implemented, and any extra features
  - This will be viewed during or before the viva, so should be a clear explanation of your work
- Viva: after the video explanation, you are required to give a demo of your work, and answer any questions
  - This includes making changes as requested and running the code for a live demo
  - Other questions that may be unrelated to the project itself but are relevant for the course

Instructions

- This is a live document and will be updated with more details and FAQs (possibly including suggested wireframes, but not specific implementation details) as we proceed.
- We will freeze the problem statement on or before 24th September, beyond which any modifications to the statement will be communicated via proper announcements.
- The project has to be submitted as a single zip file.