

CSC7510 Cloud and Enterprise System

## **HumorMe!**

Prepared By  
Shristi Shrestha

Prepared for  
Qingyang Wang  
15<sup>th</sup> September 2023

Table of Contents

**Introduction..... 3**  
    **Project Overview..... 3**  
    **Project Deliverables ..... 3**  
**Project Organization ..... 5**  
    **Tools and Techniques ..... 5**  
**Task Distribution ..... 6**  
    **Milestones ..... 6**  
    **Timeline..... 7**

## Introduction

### Project Overview

HumorMe! is a fun community space for people to share their sense of humor. This social media webapp aims to provide an easy-to-use public platform to post, share and rate jokes. With rise of memes and short reels in apps like Facebook and Instagram, there is a huge spike in userbase consuming exclusively funny content. With lots of conflicts and calamities happening everywhere, people need a space, even a virtual, where they can enjoy and share a moment of laugh with people of similar interests. Like the name suggests (HumorMe!), our vision to humor people and provide a sense of belonging in a digital space. In this online platform, users can post their jokes, read jokes from people they follow and actively participate in rating novel jokes.

### Project Deliverables

1. The app will be hosted in a public domain.
2. Entity-Relationship (ER) diagram identifying parameters for entities such as users, posts and comments.
3. User Service
  - a. Anyone can register using **Gmail account**. Validation function will only check for “@gmail.com” in the user email.
  - b. Logged in session will be stored in browser for easier user engagement.
4. View Jokes
  - a. Users can **search for jokes** by labels: custom and default
    - i. Default labels: lame, funny, serious and dark
    - ii. Custom labels: anything that creator assigns
  - b. Users can **search for jokes** based on the content of the joke
    - i. Use of “like” pattern matching within the context of the jokes
5. Humor content
  - a. Any logged in user can post their jokes by creating a post “Joke”.
  - b. By default, the post is labelled as “lame”.
  - c. The maximum size of each post is 150 characters including spaces.
  - d. Users can assign at most of two custom labels of their own, for example, “#doctor&patient”.
  - e. Users can hide jokes, but not delete.
  - f. Users can update jokes, which will be appended to their previous joke (there is no override of the existing joke).
6. Home Feed
  - a. Once a user is logged in, the app will display list of jokes from people the user follows. If there is no following, most popular/recent jokes will be displayed.

- b. Each joke post consists of following information
    - i. Joke content (text)
    - ii. Default and custom labels
    - iii. Text comments
    - iv. Ratings based on default labels
- 7. User profile
  - a. Users can view their profile details which consists of following information like email, display name, jokes, followers, following, %community engagement.
- 8. Community engagement
  - a. Users can rate jokes from other people using default labels (4.a.i).
  - b. Users can comment on jokes (maximum of 150 characters).

## **Project Organization**

### **Tools and Techniques**

The static web content will be stored on an EC2 instance running Apache Web Server. The web content will be created using HTML, JavaScript and jQuery, with Bootstrap CSS. Another EC2 instance running Apache Tomcat will store Java servlets (backend APIs) and MySQL server (database functionality). User created content will be persisted in the database for comparatively longer time (will be deleted after some time). Java servlets will interact with database to return dynamic web pages to the browser (client/users). Eclipse IDE will be used to develop and test Java Servlets and web content.

## Task Distribution

I have divided the overall project tasks into seven distinct milestones. Each milestone comprises of multiple functional tasks, as well as includes incremental process of writing report. By incorporating writing process, I want to distribute the load of completing report and preparing for demo throughout the project development process. In what follows, I describe tasks associated with each milestone in detail and propose a dedicate time period to achieve those milestones.

### Milestones

1. ER diagram and established relationship between the entities
  - Description: Draft an ER model with parameters for each entity. Use draw.io (free web tool) to create and show relationship among the entities.
  - Deliverables: A PDF file showing the final ER diagram for the project.
2. Static web content and templates using mock data
  - Description: Create a mock data and prepare a general look of the app without API requests. Prepare interfaces to make API calls and simply return the mock data.
  - Deliverables: A deployed app with static view of the app showing the general flow of the site.
3. User service integrated with web app
  - Description: Implement Java servlets for user registration and login. Integrate Java API with the client side (using the API interfaces as defined before in 2). Test out the implementation and save the session in the browser.
  - Deliverables: A deployed app with static view of the app and enabled user login/register.
4. Joke CRUD APIs integrated with web app
  - Description: Implement Java servlets for create, update, read and delete joke posts. Integrate Java API with the client side. Test out the integration and ensure update and delete are working as per the descriptions in Section Project Deliverables (5).
  - Deliverables: A deployed app with a partial static view of the app and enabled user login/register and joke CRUD operations.
5. Search API integrated with web app
  - Description: Implement Java servlets for search joke posts by labels and content keywords. Integrate Java API with the client side. Test out the integration.
  - Deliverables: A deployed app with partial static view of the app, enabled user login/register, joke CRUD operations and search jokes by query keywords.
6. Comment and rate APIs integrated with web app
  - Description: Implement Java servlets for rating and commenting on joke posts. Integrate Java API with the client side. Test out the integration.

- Deliverables: A deployed app with enabled user login/register, joke CRUD operations, search jokes by query keywords and forms to submit joke ratings/comments. Ensure that the comments/ratings are reflected in each joke post both in number statistics and threads (original joke followed by comments). Test out the whole APIs integration.
7. Complete project report and prepared in-class demo
    - Test out the whole project using public domain address
    - Refactor project report
    - Prepare a demo video and in-class presentation

### **Timeline**

1. Up to Sept 19: Finish milestones 1 and 2
2. September 20-24: Finish milestones up to 4
3. September 25-30: Finish milestones 5, 6 and 7
4. Up to October 5: Finish if anything is remaining, or needs fixing and complete documentation