TE-Gram Project

Our entire IT Team had the good fortune of hitting on a huge group PowerBall lottery ticket. Unforntunately for us they has all resigned.

They were in the middle of a very important project that has a deadline of Friday night and now we have no one to complete it.

You have been engaged to complete the unfinished portions of the the project by the deadline.

Project Overview

TE-Gram is a photo-sharing app that will allow users to upload photos that can be viewed by anyone with access to the TE-Gram site.

Users can browse photos, indicate which ones they like, bookmark their favorites, display all photos from a particular contributer, view all their favorites and upload new photos.

Current status:

- Items that are know to exist in some form:
    - A repository and directory structure to hold project assets.
    - Several tables and associated DAOs
    - Wireframes for web pages showing expected layout and content
    - A home/landing page
- Requirements still to be implemented:
    - As a user, I would like to be able to add new comments to an existing photo.
    - As a user. I would like to tap/click on a contributor name and see a list of photos submitted by the contributor.
    - As a user, I would like to tap/click an indictor to show I like a particular photo.
    - As a user, I would like to bookmark a photo as a favorite.
    - As a user, I would like to be able to list all my favorites by user.
    - As a user, I would like to be able to add new photos.

Collaborating in a shared repository

Each team will be working in the same shared repository. In order to collaborate in this manner, each team will need to:

1. Clone the shared repository.
2. Create a branch for their work.
    - `cd to cloned repository directory`
    - `git branch new-branch-name`
3. Checkout the branch for their work.

- `git checkout new-branch-name`

Note: the `git branch` command can be used to see which branches you have checked out locally. The current working branch will be shown with an '*' next to it's name.

4. Perform all their work in their branch.

5. When ready to save changes to Bitbucket branch:

    - `git add -A`
    - `git commit -m'commit message'`
    - `git push origin new-branch-name`

6. To get your change reviewed and merged in to the master branch:

    - Go to the repository in Bitbucket
    - Click the plus sign on the left side and choose `Create a pull request` You can see your branch as the source branch and master in the destination branch.
    - `git commit -m'commit message'`
    - Be sure to provide a meaningful description in the Pull Request and select at least 3 other team members (along with the instructors) as reviewers.

Running TE-Gram (Java)

Backend

1. Open terminal, navigate to `~/workspaces/.../module-4/Sample_Projects/instagram/backend/java/TEGram`, and execute the following:

```
$ dbcreate -U postgres TEGram
$ psql -U postgres -d TEGram < database/scheme.sql
$ psql -U postgres -d TEGram < database/data.sql
```

2. Open Eclipse and import `~/workspaces/.../module-4/Sample_Projects/instagram/backend/java/TEGram`.

3. Navigate to "Servers" view, right-click on "Tomcat v9 Server at localhost", remove all but "TEGram(TEGram-0.0.1-SNAPSHOT)" from the configured panel.

4. Right-click on the TEGram project in the Project Explorer and "Run As Run On Server" There isn't any additional configuration on the backend. Once the database has been created, and project imported, TEGram should just run.

Frontend

5. Back at the terminal, navigate to `~/workspaces/.../module-4/Sample_Projects/instagram/frontend`, and edit `.env` to point to the instance of TEGram Tomcat server running under Eclipse.

```
$ code .env
```

Change the one-and-only line to "VUE_APP_REMOTE_API=http://localhost:8080/TEGram"

6. Install the Vue front end

```
$ npm Install
```

7. Run the Vue app

```
$ npm run serve
```

8. Open `http:localhost:8081/` in Chrome With any luck, the hope page of TEGram will be displayed. :smiley: