JavaScript internship – Instagram

Welcome to JavaScript internship in Levi9! Let’s build a lite version of the most popular social network today – Instagram. The assignment will include both backend (Nodejs, MongoDB) and frontend (Angular SPA).

Description

User performs login.  
After successful login, user is transferred to home page where recent posts of users that are followed are shown in chronological order.  
User can click on a post to open it in preview mode, where comments can be viewed and posted.  
User can search for other users by name.  
User can open own and other user’s profiles. Profile contains user name, avatar, list of posts and follow/unfollow button if profile is from other user.  
User can follow or unfollow other users.  
User can create new post by uploading image and entering post text.

Important notes:

Implementation of user registration is not needed, list of users will be provided in JSON file.  
All user profiles are “public”, no privacy control here.  
All users will have the same password for easier testing.

Technical requirements and guide

Database – MongoDB

Models:  
- **User**: {string} id, {string} name, {string} imageUrl, {userId[]}  
- **Post**: {string} id, {userId} userId, {string} imageUrl, {string} description, {date} created, {**Comment**[]} comments  
- **Comment**: {string} id, {userId} authorId, {string} text, {date} created

Collections:  
- Users  
- Posts

Backend – Nodejs

Endpoints:

- ~~login page~~  
- home page: list of posts from users that are followed, ordered by most recent  
- single post page: image and text, with list of comments  
- new post page: upload image and enter text  
- profile page: user name, avatar, list of posts and information if user is followed or not  
- search users page: list of all users, or filtered by name  
- login (from login page)  
- new post (from new post page)  
- follow / unfollow user (from profile page)  
- new comment on post (from single post page)

Frontend – Angular

Routes:

- login  
- home  
- new post  
- single post  
- profile  
- search

*\* All pages*

All pages should have header, where fancy logo is on left side and on right side are links: Home, My Profile, Logout.

*Login*

Form, centered on page, username and password inputs, login button. Validate if fields are not empty.

*Home*

List of posts from users that are followed, ordered by most recent.  
Post component has header and image.  
Header has avatar and name of post author on the left, created date on the right.  
Image is below.

*New post*

Input field to choose image file, input field to enter post text. Validate if file is chosen and if it is image type.

*Single post*

Post component is the same as on Home page.  
Comments are below the image, from oldest to newest.  
New comment form is below, has input field and submit button. Validate if input is not empty.

*Profile*

Profile has avatar and name on the left.  
If profile is now by logged in user, it should show follow / unfollow button on the right.  
Below is list of posts, ordered by most recent. Post component is the same as on Home page.

*Search*

Search has input field at the top where user name can be entered. Search should be partial.  
Below is list of users that match the search query.  
User component has avatar and user name on the left. On right there is “view profile” button.

Bonus

If we have time, here are some additional requests:

- implement post text search  
- implement hashtag support: identify hashtags in post text when new post is created, linkify them when post is shown on any page, click on hashtag should list all posts that have that hashtag – reuse post search functionality!

Courses

<https://www.udemy.com/the-complete-nodejs-developer-course-2/>

<https://www.udemy.com/the-complete-guide-to-angular-2/>