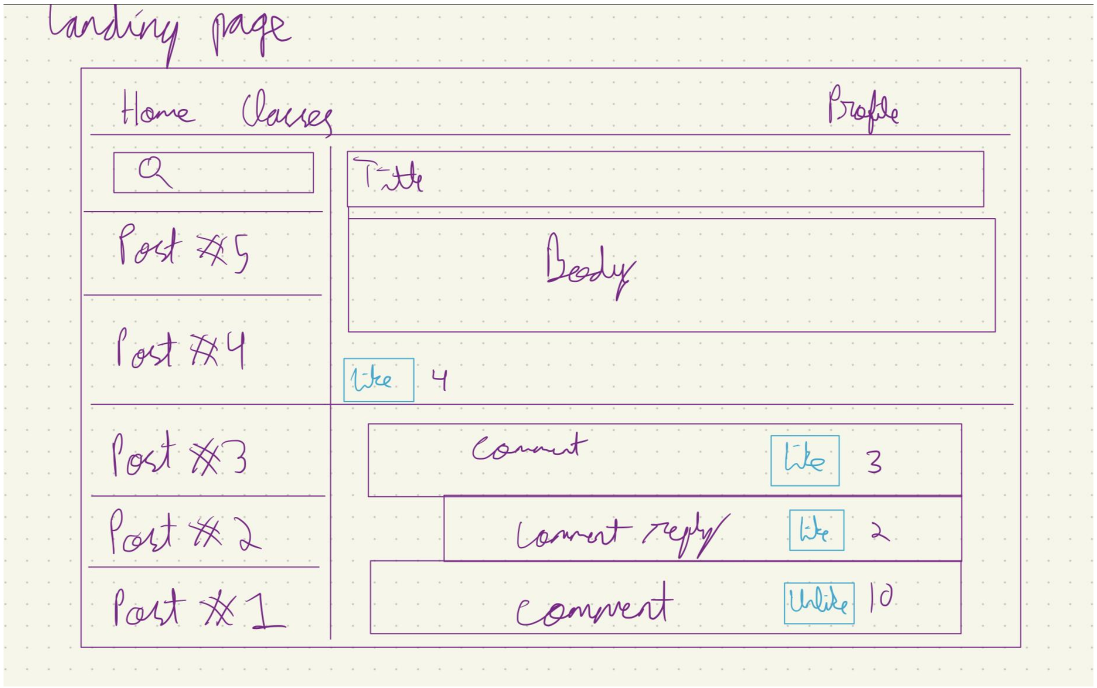
Form Application Design Documentation

Sample Screens



Note:

DB = Database

FE = Frontend

BE = Backend

TBD = To be determined

To see exact json specification, go to <domain_name>/api/schema/swagger-ui/

Experimentation needed with user authentication api points (hence TBD notes)

API Endpoints w/ Intended Function

- **GET** ***/api/schema/***
    - Functionality: Produce an up-to-date version of the current api endpoints

  - FE Status: Done (Spectacular library in Django)
- **GET** ***/api/v1/comment-likes/***
  - Functionality: Return json data of all likes
  - FE Status: TODO
  - FE implementation: Use filter function on FE side to calculate total #likes
- **POST** ***/api/v1/comment-likes/***
  - Functionality: Create like and update it to database
  - FE Status: TODO
  - FE implementation: A like button that sends this post request
- **DELETE** ***/api/v1/comment-likes/{id}/***
  - Functionality: Unlike by deleting the posted like by user
  - FE Status: TODO
  - FE implementation: An unlike button that sends this delete request
- **GET** ***/api/v1/comment-reply-likes/***
  - Functionality: Retrieve all likes for a comment reply
  - FE Status: TODO
  - FE implementation: Filtered by FE to determine the # likes for comment reply
- **POST** ***/api/v1/comment-reply-likes/***
  - Functionality: Post a reply like to database
  - FE Status: TODO
  - FE implementation: A like button for the comment reply that sends request
- **DELETE** ***/api/v1/comment-reply-likes/{id}***
  - Functionality: Delete comment reply like (unlike)
  - FE Status: TODO
  - FE implementation: Unlike button for comment reply likes
- **GET** ***/api/v1/comment-replys/***

    - Functionality: Return all comment replies in DB
    - FE Status: TODO
    - FE implementation: Filter comment replies based on class/query on FE
- **POST** ***/api/v1/comment-replys/***
    - Functionality: Post a comment reply to DB
    - FE Status: TODO
    - FE implementation: Create “reply to comment” button for each comment
- **GET** ***/api/v1/comment-replys/{comment-reply-id}/***
    - Functionality: Retrieve a specific comment
    - FE Status: TODO
    - FE implementation: TBD
- **PATCH** ***/api/v1/comment-replys/{comment-reply-id}/***
    - Functionality: Edit the text of comment reply
    - FE Status: TODO
    - FE implementation: Edit/submit button corresponding to sending patch request
- **DELETE** ***/api/v1/comment-replys/{comment-reply-id}/***
    - Functionality: Delete comment reply from DB
    - FE Status: TODO
    - FE implementation: Delete button that sends this request
- **GET** ***/api/v1/comments/***
    - Functionality: Retrieve list of all comments
    - FE Status: TODO
    - FE implementation: Filter on FE by query or class/subject etc
- **POST** ***/api/v1/comments/***
    - Functionality: Send post request for new comment for a post (post id is specified within json body of the post request)

  - FE Status: TODO
  - FE implementation: create comment button for a post that sends request
- **GET *api/v1/comments/{comment-id}/***
  - Functionality: Send a get request for a specific comment
  - FE Status: TODO
  - FE implementation: TBD
- **PATCH */api/v1/comments/{comment-id}/***
  - Functionality: Edit a comment to a post to DB
  - FE Status: TODO
  - FE implementation: A post comment button that does the request
- **DELETE */api/v1/comments/{comment-id}/***
  - Functionality: Remove a comment from a post
  - FE Status: TODO
  - FE implementation: A delete comment button
- **POST */api/v1/dj-rest-auth/logout/***
  - Functionality: A post request for logging out
  - FE Status: TODO
  - FE implementation: A logout button that sends request
- **POST */api/v1/dj-rest-auth/lgoin/***
  - Functionality: A post request for logging in where user enters username and password. Returns a token via JSON to be stored in future HTTP headers for accessing various APIs.
  - FE Status: TODO
  - FE implementation: A login button that sends request
- **POST */api/v1/dj-rest-auth/registration/***

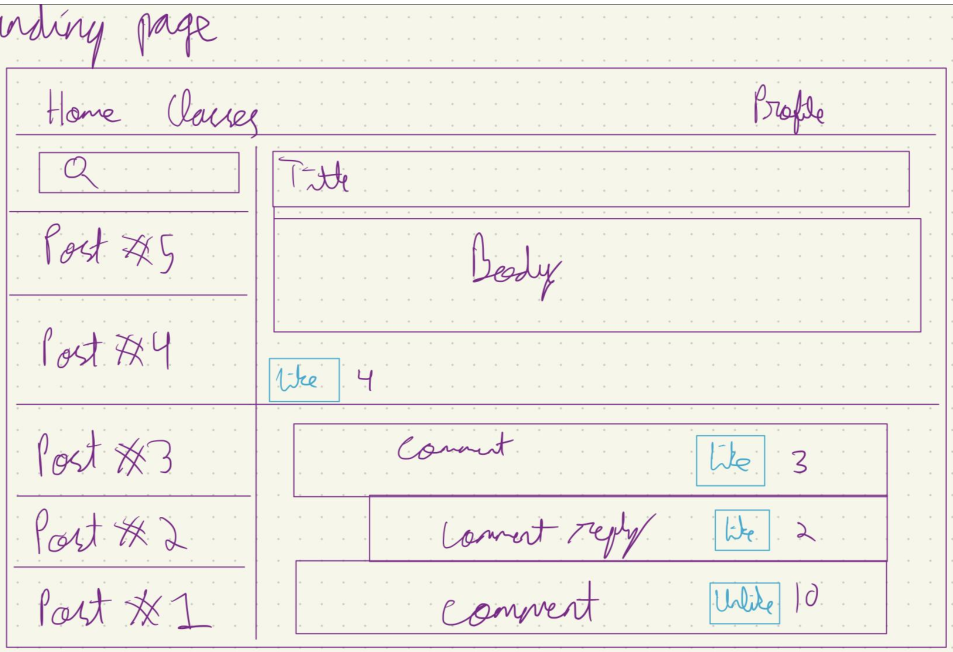
    - Functionality: Register a new user with username, email, password1 and password2 (redundancy to ensure user remembers password) through a post request.
    - FE Status: TODO
    - FE implementation: Modern UI form that makes post request
- **POST** ***/api/v1/dj-rest-auth/password/reset/***
    - Functionality: Reset the password of a user, post request that contains email, sends token to user’s email
    - FE Status: TODO
    - FE implementation: TBD
- **POST** ***/api/v1/dj-rest-auth/password/reset/confirm***
    - Functionality: Allows users to change their password. The parameters are password1, password2, uid, and token. **Unsure what UID is supposed to be.**
    - FE Status: TODO
    - FE implementation: Modern UI Form that submits post request
- **POST** ***/api/v1/dj-rest-auth/registration/resend-email/***
    - Functionality: TBD
    - FE Status: TODO
    - FE implementation: TBD
- **POST** ***/api/v1/dj-rest-auth/registration/verify-email/***
    - Functionality: TBD
    - FE Status: TODO
    - FE implementation: TBD
- **GET** ***/api/v1/post-likes/***
    - Functionality: Retrieve list of all post likes

  - FE Status: TODO
  - FE implementation: Retrieve when loading a post, filter based on post-id
- **POST** ***/api/v1/post-likes/***
  - Functionality: Create a like for a post
  - FE Status: TODO
  - FE implementation: Like post button
- **DELETE** ***/api/v1/post-likes/{post-like-id}/***
  - Functionality: Delete the like of post
  - FE Status: TODO
  - FE implementation: Unlike button
- **GET** ***/api/v1/posts/***
  - Functionality: Retrieve list of all posts
  - FE Status: TODO
  - FE implementation: Filter on FE by query or class/subject etc
- **POST** ***/api/v1/posts/***
  - Functionality: Send post request for a post
  - FE Status: TODO
  - FE implementation: create post button for a post that sends request
- **GET** ***/api/v1/posts/{post-id}/***
  - Functionality: Send a get request for a specific post
  - FE Status: TODO
  - FE implementation: TBD
- **PATCH** ***/api/v1/posts/{post-id}/***
  - Functionality: Modify an existing post
  - FE Status: TODO
  - FE implementation: An edit post button

- **DELETE** ***/api/v1/posts/{post-id}/***
  - Functionality: Remove a post
  - FE Status: TODO
  - FE implementation: A delete post button
- **GET** ***/api/v1/users/***
  - Functionality: Retrieve list of all users
  - FE Status: TODO
  - FE implementation: TBD
- **POST** ***/api/v1/users/***
  - Functionality: TBD
  - FE Status: TODO
  - FE implementation: TBD
- **GET** ***/api/v1/users/{user-id}/***
  - Functionality: Send a get request for a specific post
  - FE Status: TODO
  - FE implementation: TBD
- **PATCH** ***/api/v1/users/{user-id}/***
  - Functionality: Modify an existing user
  - FE Status: TODO
  - FE implementation: An edit user field minus password settings
- **DELETE** ***/api/v1/users/{user-id}/***
  - Functionality: Remove a user
  - FE Status: TODO
  - FE implementation: A delete user account permanently

Sample Screens

Landing page


Home
Classes
Profile
Title
Post #5
Body
Post #4
Like
4
Post #3
Comment
Like
3
Post #2
Comment reply
Like
2
Post #1
comment
Unlike
10