INTROVERT HUB MVP SPECIFICATION

This application will have five vital components/features: User registration/authentication, User posts, Search users, Like and Comment on user posts, and Follow/Unfollow users.

- **User registration/Authentication:** To access all the other features of the introvert hub, the user must first register on the application. We will collect users' fundamental data such as email, username, first name, last name, and password, and store it in the database. However, users can change their password for security purposes.

- **User profiles:** Users will be able to create and edit their profiles, including their name, profile picture, bio, email address, and username. The altered data reflects in the database system and the front end of the application.

- **User posts:** Users will be able to have access to posts and modify them accordingly and can as well post photos that show and see updates from other users they follow and be able to post their own updates.

- **Follow/Unfollow users:** The logged-in user can follow/unfollow users from the user suggestion section.

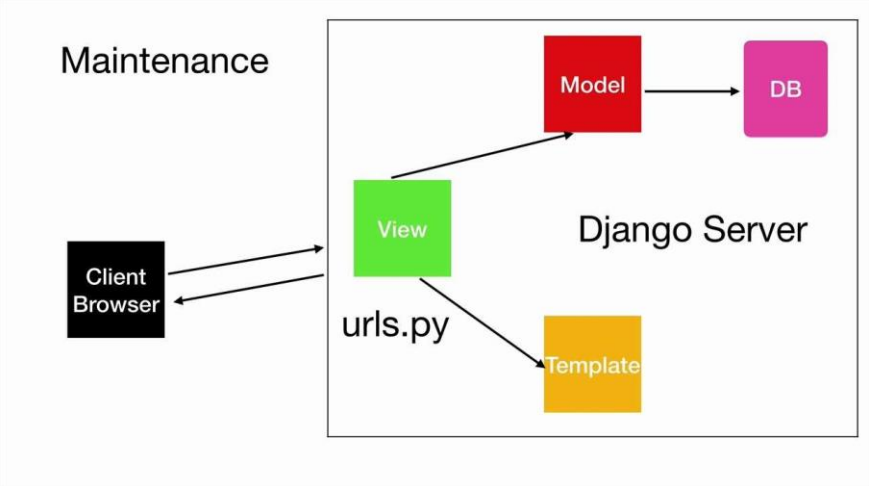
- **Search users:** Users can search for other users by entering the username in the search bar provided in the home page. A list of all users matching the username typed will appear as a search result.

The above features will help create a basic, functioning MVP for the Introvert Hub project. Future iterations can incorporate additional features based on user feedback, such as the ability to share files or collaborate on projects.

ARCHITECTURE

The project will be developed using an iterative and incremental approach. The first step will be to define the requirements and design the system architecture. This will involve identifying the various components of the system and their interactions, as well as defining the user flows and functionalities.

Once the requirements and design are finalized, the implementation phase will begin. This will involve the development of the various components of the system, including the user interface, database, and backend logic. The system will be tested and refined throughout the development process to ensure that it meets the requirements and provides a smooth and efficient user experience.



Django is an open-source web application framework written in python. It is a high-level framework that encourages rapid development and clean, pragmatic design. Django consists of three major parts: Model, View, and Template.

APIs

Model: The model is a single, definitive data source that contains the essential field and behavior of the data. Python classes implement models. Usually, one model is one table in the database. Each Python object in the model represents a field of a table in the database. Django provides a set of automatically generated database application programming interfaces **(APIs)** for the convenience of users.

View: A view is a short form of the view file. It is a file containing a Python function that takes web requests and returns web responses. A response can be HTML content or XML documents or a “404 error” and so on. The logic inside the view function can be arbitrary if it returns the desired response. To link the view function with a particular URL, we need to use a structure called URL conf that maps URLs to view functions.

Template: Django’s template is a simple text file that can generate a text-based format like HTML and XML. The template contains variables and tags. Variables will be replaced by the result when the template is evaluated. Tags control the logic of the template. We also can modify the variables by using filters. For example, a lowercase filter can convert the variable from uppercase into lowercase. Django templates allow the developers to implement the front-end logic of the application.

Step 1 : The UI for the project was designed using HTML, CSS, SCSS, Bootstrap, and JavaScript.

Step 2: Once the frontend was designed a schema was created for different models:

Step 3: Views are written to write the underlying logic that needs to be displayed on the frontend in the views.py file and routes are configured in the urls.py file.

DATA MODELLING

User table: This table stores user data, such as their unique ID, name, email, password, and profile picture.

```
User
- id: int (primary key)
- name: varchar(255)
- email: varchar(255) (unique)
- password: varchar(255)
- profile_picture: varchar(255)
```

Profile table: This table stores profile data, such as the user's bio, location, and interests.

```
Profile
- user_id: int (foreign key references User(id))
- bio: varchar(500)
- location: varchar(255)
- interests: varchar(500)
```

Post table: This table stores feed data, such as post text, timestamp, and user ID.

```
Post
- id: int (primary key)
- user_id: int (foreign key references User(id))
- text: varchar(1000)
- created_at: datetime
```

Follower table: This table stores follow data, such as follower ID and followed ID.

```
Follower
- follower_id: int (foreign key references User(id))
- followed_id: int (foreign key references User(id))
```

Message table: This table stores message data, such as message text, timestamp, and sender/receiver ID.

```
Message
- id: int (primary key)
- sender_id: int (foreign key references User(id))
- receiver_id: int (foreign key references User(id))
- text: varchar(1000)
- created_at: datetime
```

Setting table: This table stores user setting data, such as email preferences and password.

```
Setting
- user_id: int (foreign key references User(id))
- email_preferences: varchar(255)
- password: varchar(255)
```

The data model described above stores the necessary user data and related information in separate tables to allow for efficient querying and data management. Each table contains columns that represent specific pieces of data, and each row represents a unique record of that data. This model can be adjusted and expanded upon based on the needs of the Introvert Hub project as it evolves over time.

USER STORIES

- As a user, I want to be able to create an account so that I can join the Introvert Hub community.
- As a user, I want to be able to edit my profile so that I can share information about myself with other users.
- As a user, I want to be able to post updates and share content so that I can connect with other IT introverts.
- As a user, I want to be able to follow other users so that I can see their posts and updates in my feed.
- As a user, I want to be able to search for other users based on their interests or expertise so that I can find like-minded individuals to connect with.

MOCKUPS