Socializing Platform Based on Music-Interests of User

Tanishq Chugh - PES1UG20CS460
Sundeep A - PES1UG20CS445
Jimish Mrug - PES1UG20CS180

Abstract

- Match(IT) is a socializing platform built to connect people based on their music interests.
- In this ever- evolving world, people feel more comfortable to interact with each other virtually as compared to in-person and having some sorts of similarities, makes the conversation to flow smoothly and people vibe much quicker.
- Music is one of the most important metric which binds people together as it brings out the inner sentiments and personality of a person.
- Recommending people music alike to their taste and using the same to help them connect with people who have a similar taste would be a great way to socialise.
- Hence, a product that uses Music as the metric to help people meet similar minded people can prove to be the most efficient one in its industry.

Implementation Details

Front-End:

- Front-end of the application has been developed completely in ReactJS.
- ReactJS is an open-source front-end Javascript library for building user-interfaces based on UI components.
- Various Components have been built individually and designed using CSS.

Back-End:

- Back-End of the application has been developed using NodeJS.
- It creates an eternal server which interacts with the ML-models as well as the database and performs the required user-functionality.
- Back-End interacts with ML-models to find songs similar to user's preferred song and then filters out the users whose interests are same as the recommended songs.
- Back-End interacts with Front-End using Axios,

Implementation Details

ML-Models:

Item-Based Collaborative Filtering:

- To calculate similarity in Collaborative Filtering model, Metric used is Cosine-Similarity. Cosine similarity measures the similarity between two vectors by calculating the cosine of the angle between them.
- For our model, vector ui remained constant as the user profile. And then looped through all the songs by plugging in each other user profile as Vector uj to calculate the cosine similarity between the respective songs.
- It then returns top ‘K’ songs similar to the given user’s preferred song.

DBSCAN:

- DBSCAN model is used to cluster songs in various groups based on eps (= 0.57) and min_sample (= 24).
- All the songs which belong to the cluster of user’s preferred song are returned.

Implementation Details

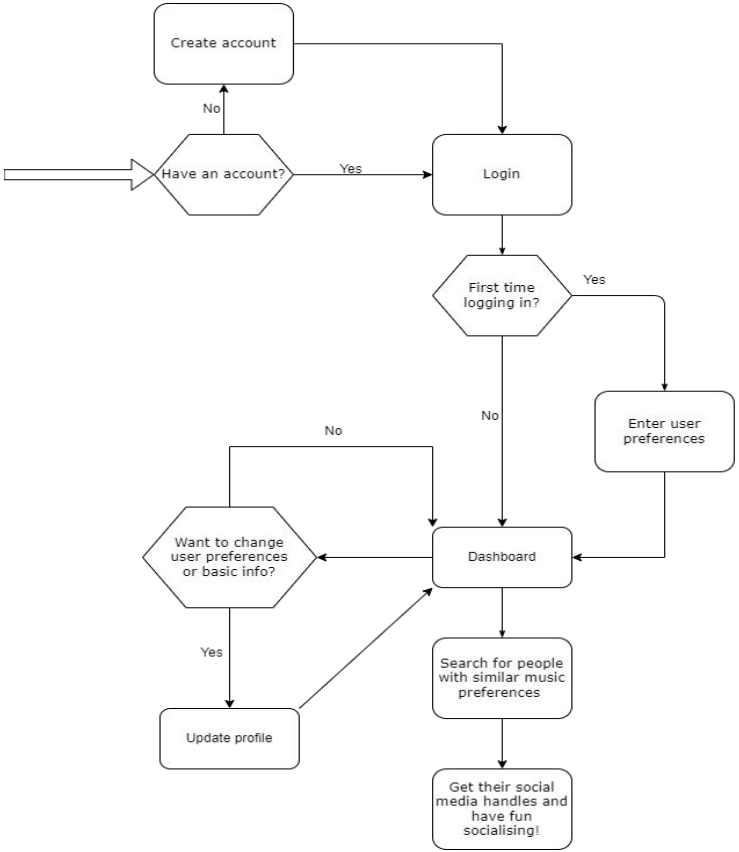
K-Nearest Neighbours:

- Ideal Number of Clusters (K) is found using the elbow method to be equal to 7.
- Taking the input as the user's preferred song, the cluster to which it belongs is found out.
- If the cluster contains more than 25 elements, 25 elements which are closest to the user's preferred song are returned, else all the songs in that particular cluster are returned.

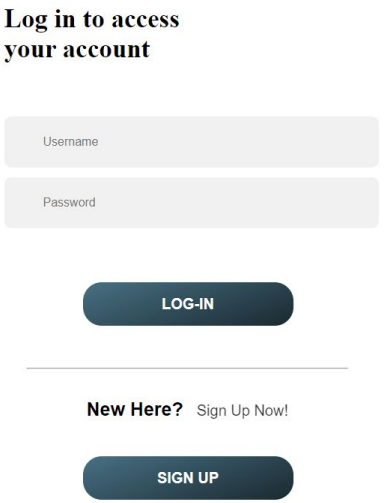
Euclidian Distance Model:

- Euclidian Distance to each song-point with respect to user's preferred song is found out.
- 500 songs closest to the user's preferred song are returned .
- This metric is used in the extreme case where no other model is able to predict users and hence this model is used as a brute-force approach.

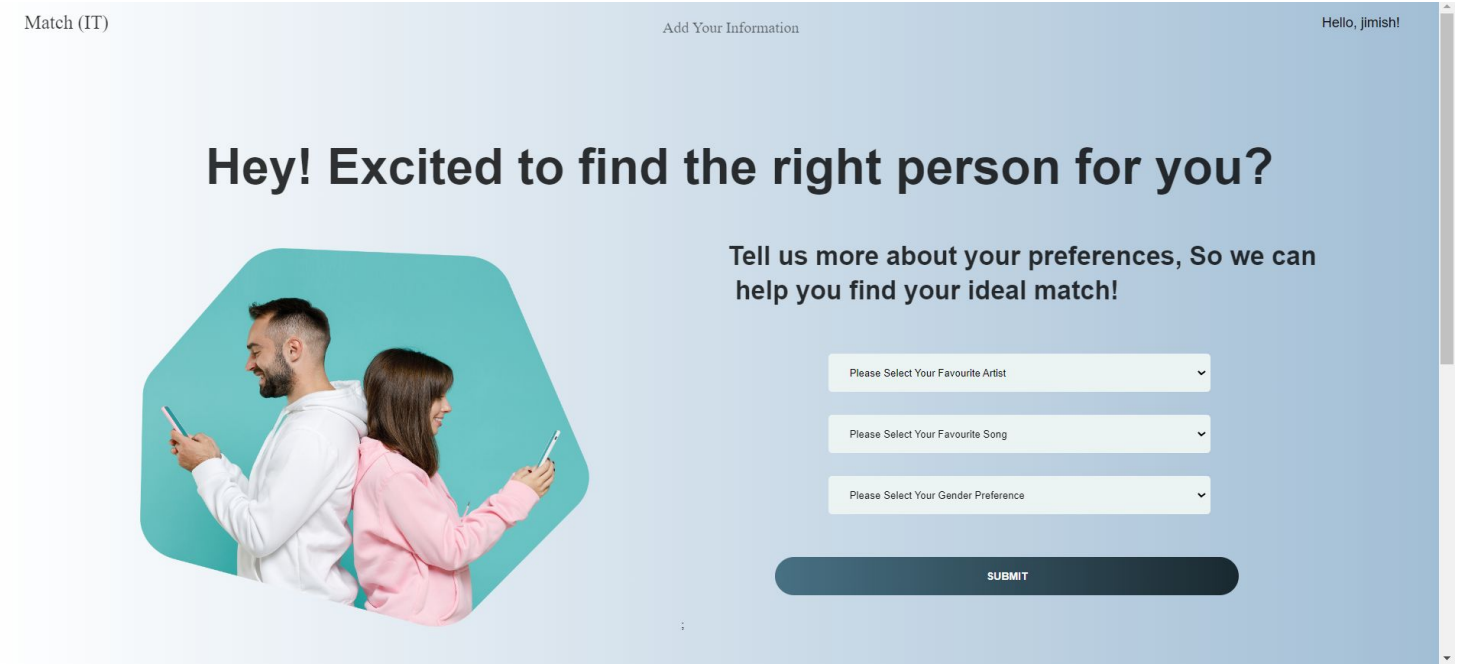
User - Flow Model



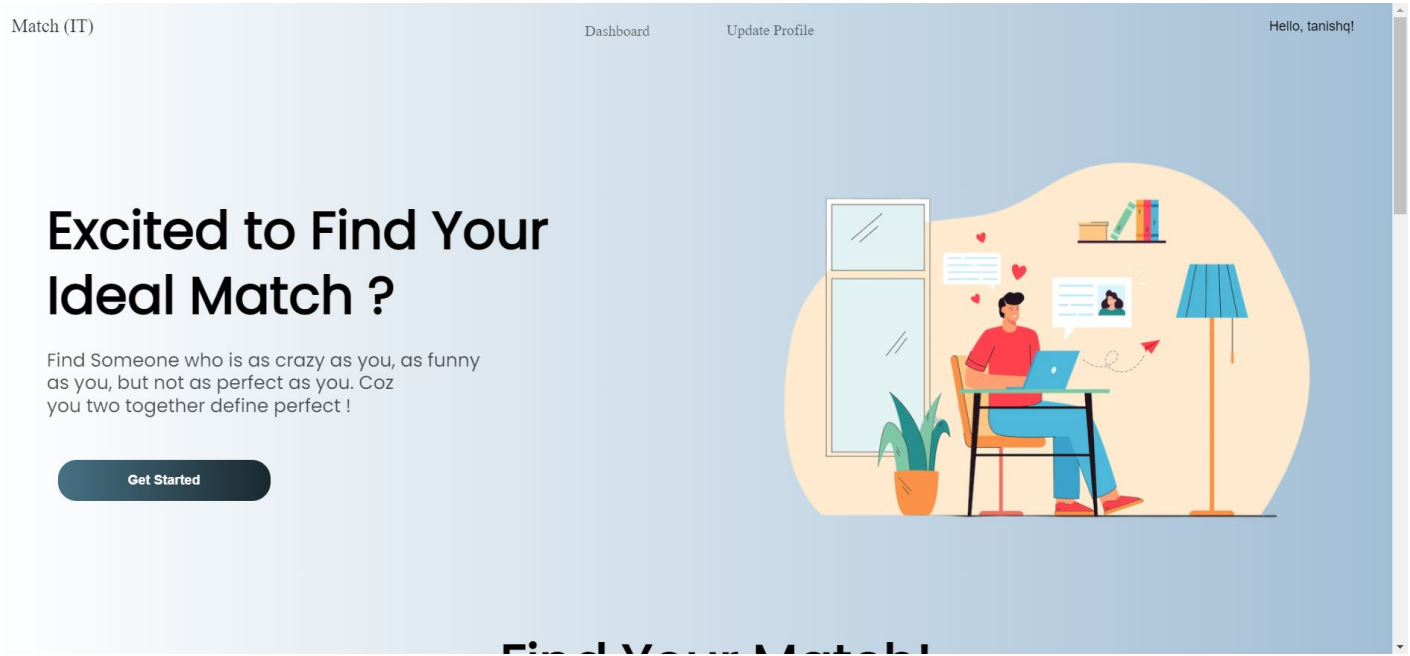
Project Demonstration



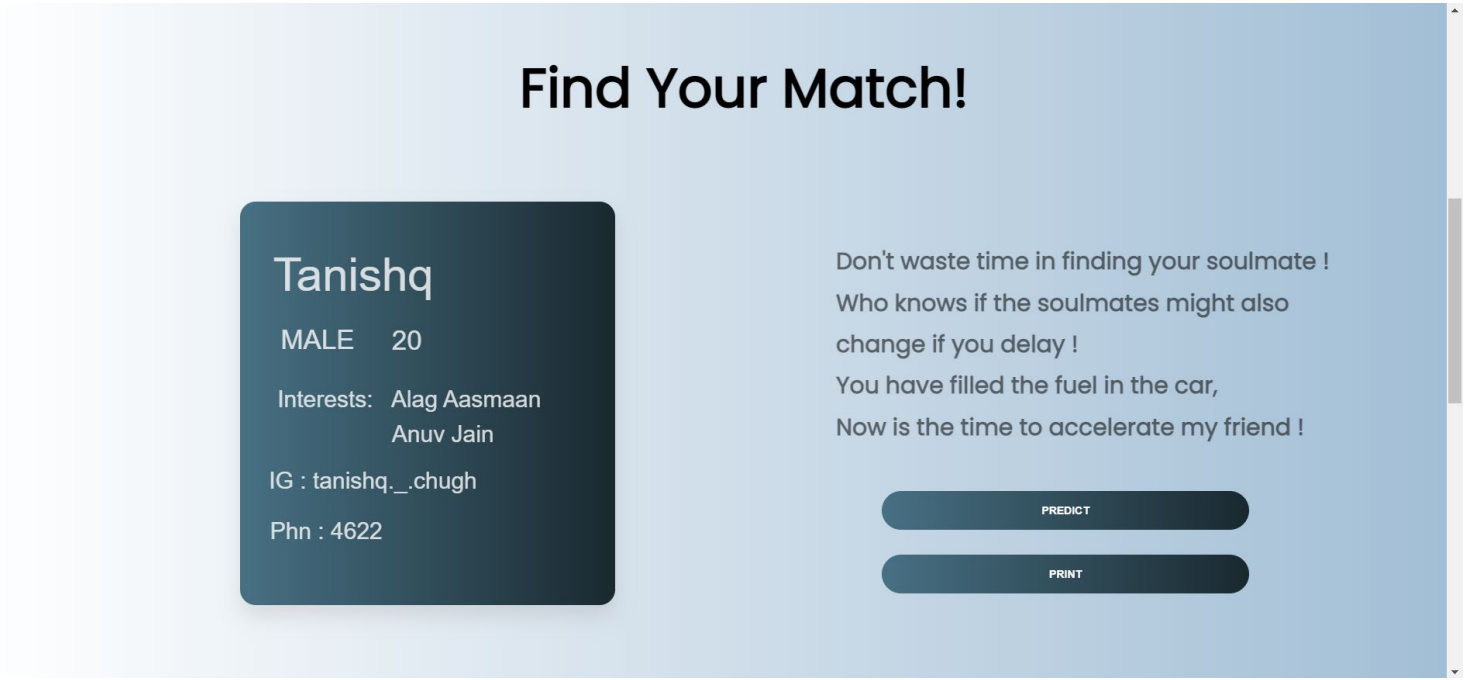
Project Demonstration



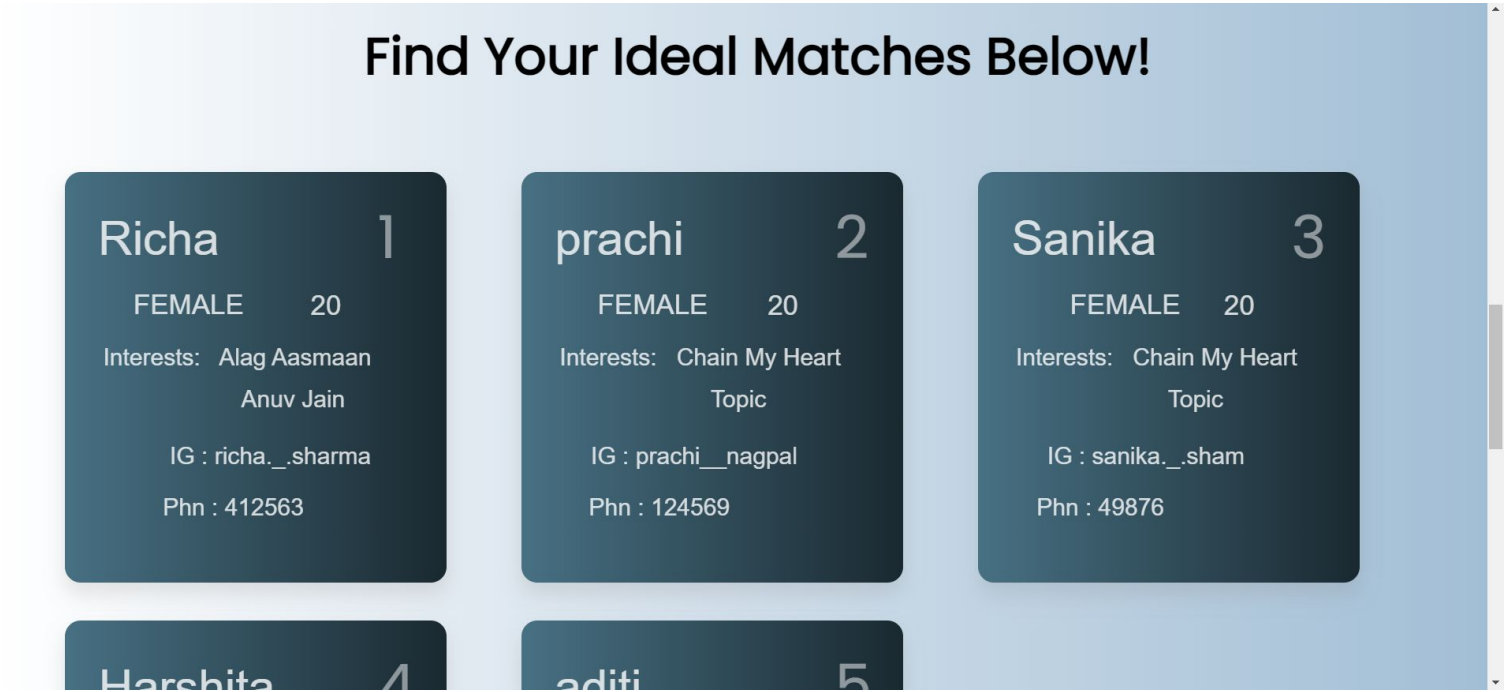
Project Demonstration



Project Demonstration



Project Demonstration



Results

With this project, we were able to make connections among various users based on their Music Taste.

Due to unavailability of datasets relating users to their music taste, we developed a multi-layered recommender system that finds similar songs with respect to user's preferred song and then matches the user to other users based on the predicted similar songs.

Models used to predict the songs are: Item-Based Collaborative Filtering (Cosine-Similarity), DBScan, K-Nearest Neighbours and Euclidian Distance.

Conclusion

We made use of multiple models to build a platform that allows the users to find and connect with people with alike music taste based on the preferences entered by the user.

We provide users flexibility to update their preferences so that they can choose to find and get in touch with people according to their current music taste and not be fixed onto one which enables the user to interact with people of various music taste.

We believe this platform is effectively serving the purpose of helping users socialise in a unique and a more practical approach as well.

Future Scope

- Deployment & Testing of the product.
- Developing an interface to take feedback from users to develop a dataset relating users to his/her music interests as well as testing the application based on match-predictions.
- Develop this application to be a fully functional social network including features like - chat interface, user posts, etc.

Thank You!