How to install?

Prerequisites:

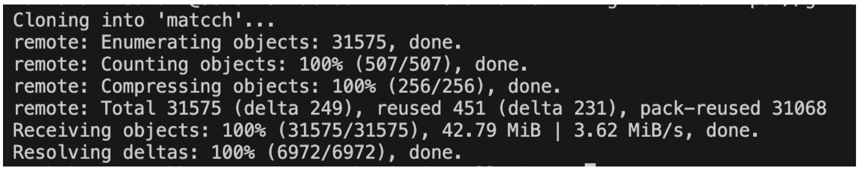
1. Git
2. Docker

Installation

1. Make sure that docker and docker compose are installed and running.

2. Clone the repository using:

    `git clone https://github.com/sumukh03/matcch.git`



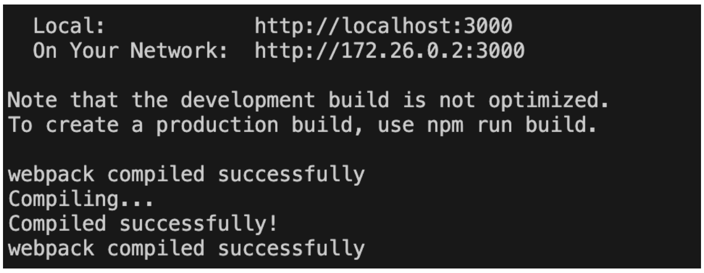
```
Cloning into 'matcch'...
remote: Enumerating objects: 31575, done.
remote: Counting objects: 100% (507/507), done.
remote: Compressing objects: 100% (256/256), done.
remote: Total 31575 (delta 249), reused 451 (delta 231), pack-reused 31068
Receiving objects: 100% (31575/31575), 42.79 MiB | 3.62 MiB/s, done.
Resolving deltas: 100% (6972/6972), done.
```

3. Move to the working directory using the command:

    `cd matcch`

4. To start the application run the command:

    `docker compose up --build`



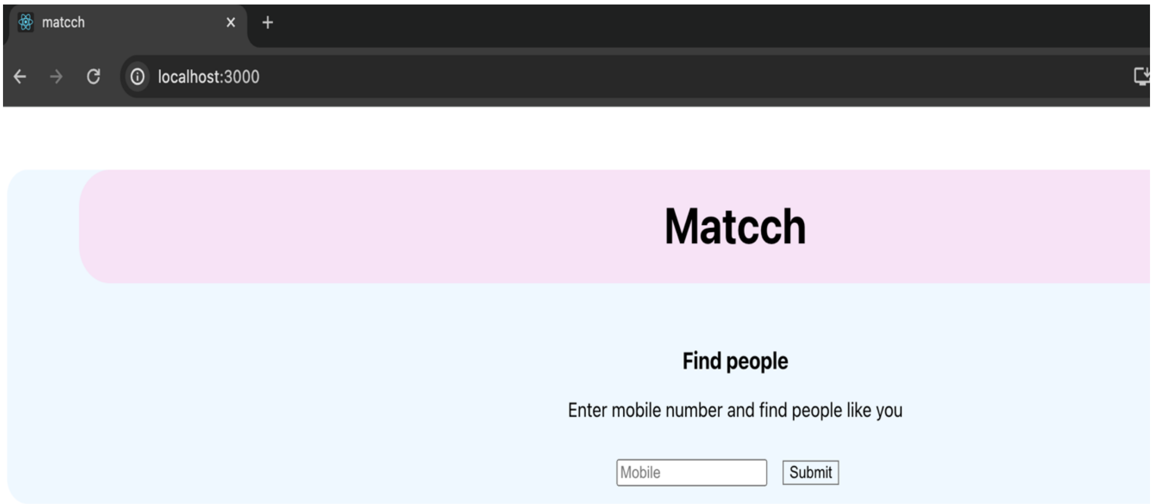
```
  Local:            http://localhost:3000
  On Your Network:  http://172.26.0.2:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Compiling...
Compiled successfully!
webpack compiled successfully
```

5. To access the application, inside the BROWSER open

http://localhost:3000/



Matcch

Find people

Enter mobile number and find people like you

6. Refer How to use the application further

Using the application

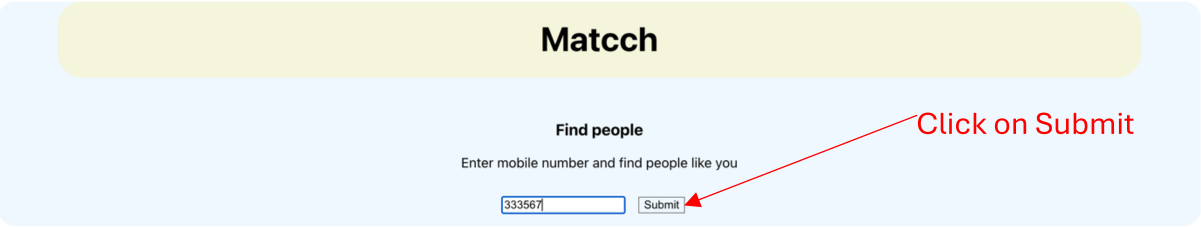
After successfully installing the application, and accessing it on localhost:3000

The user interface has 3 pages

1. The Entry Page
2. The Questions Page
3. The Recommendation Page

The Entry Page

- The entry page consists of an input box
- The user must enter the mobile number
- Click on submit



Matcch

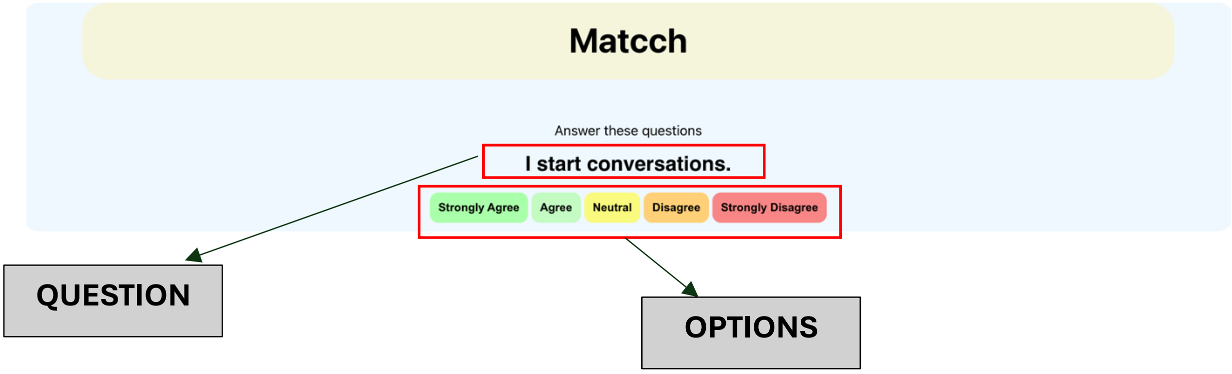
Find people

Enter mobile number and find people like you

333567

The Question Page

- After the user enters the mobile number and submits
- We have the Question page where the user is supposed to answer the questions
- Every question contains 5 options, the user must choose one option that suits the best about them

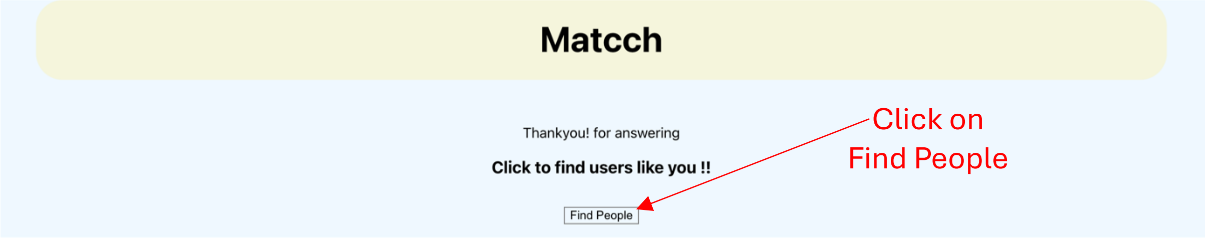


Matcch

Answer these questions

I start conversations.

- After answering all the questions, click on Find People Button



Matcch

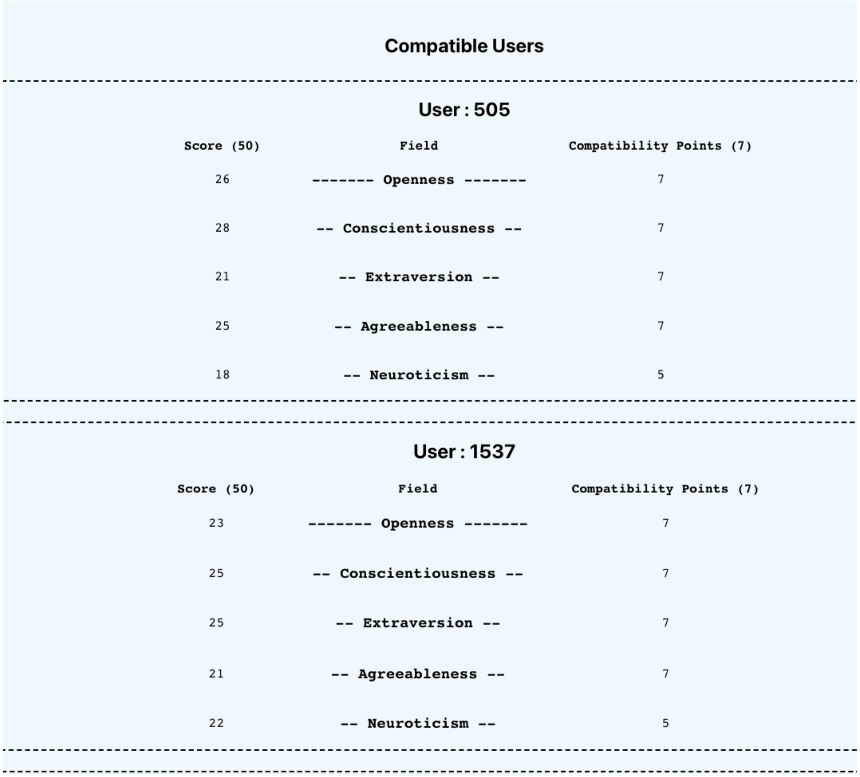
Thankyou! for answering

Click to find users like you !!

Click on Find People

The Recommendation Page

- The recommendation page shows the **user ids**, the **scores** that the users got in the test and the **compatibility points** between you and the users in the database.



Compatible Users

User : 505

Score (50)	Field	Compatibility Points (7)
26	------- Openness -------	7
28	-- Conscientiousness --	7
21	-- Extraversion --	7
25	-- Agreeableness --	7
18	-- Neuroticism --	5

User : 1537

Score (50)	Field	Compatibility Points (7)
23	------- Openness -------	7
25	-- Conscientiousness --	7
25	-- Extraversion --	7
21	-- Agreeableness --	7
22	-- Neuroticism --	5

- These users are the most compatible users according to your personality.

APIs

- The backend application can also be used using API’s
- The below are the APIs available.
- Please find the Postman collection in `matcch/docs`

Name of API	Request Type	URL	Data (sample)	Response (sample)	Description
details	POST	http://127.0.0.1:5000/users_end/details	{ **"mobile"**: "5001" }	{ **"data"**: 5002, **"message"**: "User Created", **"status"**: true }	• Used to create new user • Data is the user_id of the new user • Status represents successful operation
test_score	POST	http://127.0.0.1:5000/process_end/test_score	{"**answers**": ["5", "1", "5", "5", "5", "5", "4", "4", "4", "4", "4", "4", "4", "4", "4", "4", "4", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3"], "**user_id**":5002 }	{ **"data"**: { "505": [[26, 28, 21, 25, 18], [7, 7, 7, 7, 5]], "1537": [[23, 25, 25, 21, 22], [7, 7, 7, 7, 5]], "1781": [[23, 26, 28, 26, 22], [7, 7, 7, 7, 5]], "2049": [[22, 21, 28, 21, 29], [7, 7, 7, 7, 5]], "3497": [[30, 23, 22, 22, 19], [5, 7, 7, 7, 5]] }, **"message"**: "compatibility recommendations ", **"status"**: true }	• Used to post the test score with the user_id • Returns the most compatible users along with the scores and points
recommendations	POST	http://127.0.0.1:5000/users_end/recommendations	{**"user_id"**:5002}	{ **"data"**: { "678": [[31, 32, 36, 43, 34], [7, 7, 7, 7, 7]], "3260": [[37, 35, 34, 41, 44], [7, 7, 7, 7, 3]], "3608": [[28, 35, 37, 39, 41], [7, 7, 7, 5, 3]], "4379": [[32, 30, 33, 39, 42], [7, 7, 7, 5, 3]], "4635": [[34, 38, 30, 50, 42], [7, 7, 7, 7, 3]] }, **"message"**: "compatibility recommendations ", **"status"**: true }	• Takes the user_id • Returns the most compatible users along with the scores and points • If the user does not exist, returns false