

Technical Challenge - Frontend (Flutter/Nextjs):

Create a mobile app (Flutter App) - Open the figma link below

- Create the Flutter app based on design.
- Implement BLOC in Flutter or Implement routes.
- Writing unit tests, widget tests & integration tests.
- Usage of good design patterns & architecture patterns.
- Build and Connect with API

Or

Create Mobile Webapp in Nextjs - Open the figma link below:

- Create using Nextjs 13 with app router based on design.
- Implement tailwind css custom and modular configuration
- Usage of good design patterns & react architecture patterns.
- Build and Connect with API

Attachments

1. Design :

<https://www.figma.com/file/VnqmoYfwdTzN8qvvDZn6GC/YouApp-Test?node-id=0%3A1&=p7hNpbhefNuFtLs7-0>

2. API : <http://techtest.youapp.ai/>

API Docs: <https://techtest.youapp.ai/api-docs#/>

a. Login

url : /api/login

b. Register

url : /api/register

c. Get Profile

url : /api/profile

d. Update Profile

url : /api/profile

(Feel free to mock the api if it is down/offline)

Zodiac/Horoscope Calculation:

<https://docs.google.com/spreadsheets/d/1Oahej8yuEHfDsQI-AwycEpQ0CnjKMsxOMq2ywMKnjsq/edit#gid=1538893505>

Technical Challenge - Backend:

1. Create a project of login/profile/chat using Nest.Js, MongoDB and Node.js implemented using docker with JWT Token, DTO, validations, socket.io or rabbit mq and unit tests.
2. Visit our figma:
<https://www.figma.com/file/VnqmoYfwdTzN8qvvdZn6GC/YouApp-Test?node-id=0%3A1&=p7hNpbhefNuFtLs7-0> and extract the fields needed to build the database schema in MongoDB and API in Nest.js for the login/profile. For the chat room, do as you prefer with as many features that shows off your knowledge of object oriented programming and data structures. Please do the chat with more effort as we will grade it more thoroughly.
3. Do **CRUD** with as many features that shows off your knowledge of object oriented programming and data structures. Show off your knowledge of schema planning and micro services in NOSQL. You will build the above figma of login and profile (Take note of the horoscope and zodiac in the profile) as well as text based chat between user A and user B with **RabbitMQ** & send a message to notify each user whenever a message is received.
4. Document and create API by adding the params below so that any frontend developers may use it.
 - a. Register
url : /api/register
 - b. Login
url : /api/login
 - c. Create Profile
url : /api/createProfile
 - d. Get Profile
url : /api/getProfile
 - e. Update Profile
url : /api/updateProfile

f. View Messages

url : /api/viewMessages

g. Send Message

url : /api/sendMessage

5. You will be judged for your ability to build the most complete backend with validations, **documentation and the best practices for api & database.**

For Reference

Zodiac/Horoscope Calculation:

<https://docs.google.com/spreadsheets/d/1Oahej8yuEHfDsQI-AwycEpQ0CnjMxOMg2ywMKnjsg/edit#gid=1538893505>

Popular types of Data Structures:

- *Array*
- *Linked List*
- *Stack*
- *Queue*
- *Binary Tree*
- *Binary Search Tree*
- *Heap*
- *Hashing*
- *Graph*
- *Matrix*
- *Other advanced data structures*

Technical Challenge - Fullstack or Lead Developer:

Do both Frontend and Backend.