Milestone 3

Studiando

Names:

1-Omnia Karem Mahfouz | 21010299

2-Ibrahim Mohamed Elsayed | 21010023

3-Mohamed Mohamed Abdel-Moniem | 21011213

4-Mohamed Hassan Kabari | 21011115

5-Mohamed Mohamed Ibrahim | 21011211

6-Omar Hani Bishr | 21010891

- ## Introduction:
  - ### Overview and purpose of the application:

    Students often grapple with the challenge of managing multiple resources in their daily lives, leading to issues of abstraction. Our website is designed to alleviate these challenges, providing a seamless and user-friendly experience to enhance students' daily lives.

  - ### Goals and objectives:

    The goal of this project is to create a user-friendly website that combines an AI-powered search engine, Google Drive integration, an announcements section, text and voice chat features, and a collaborative to-do list. The website aims to improve communication and productivity for teams, whether they are students or professionals in various settings.

  - ### Target audience or users:

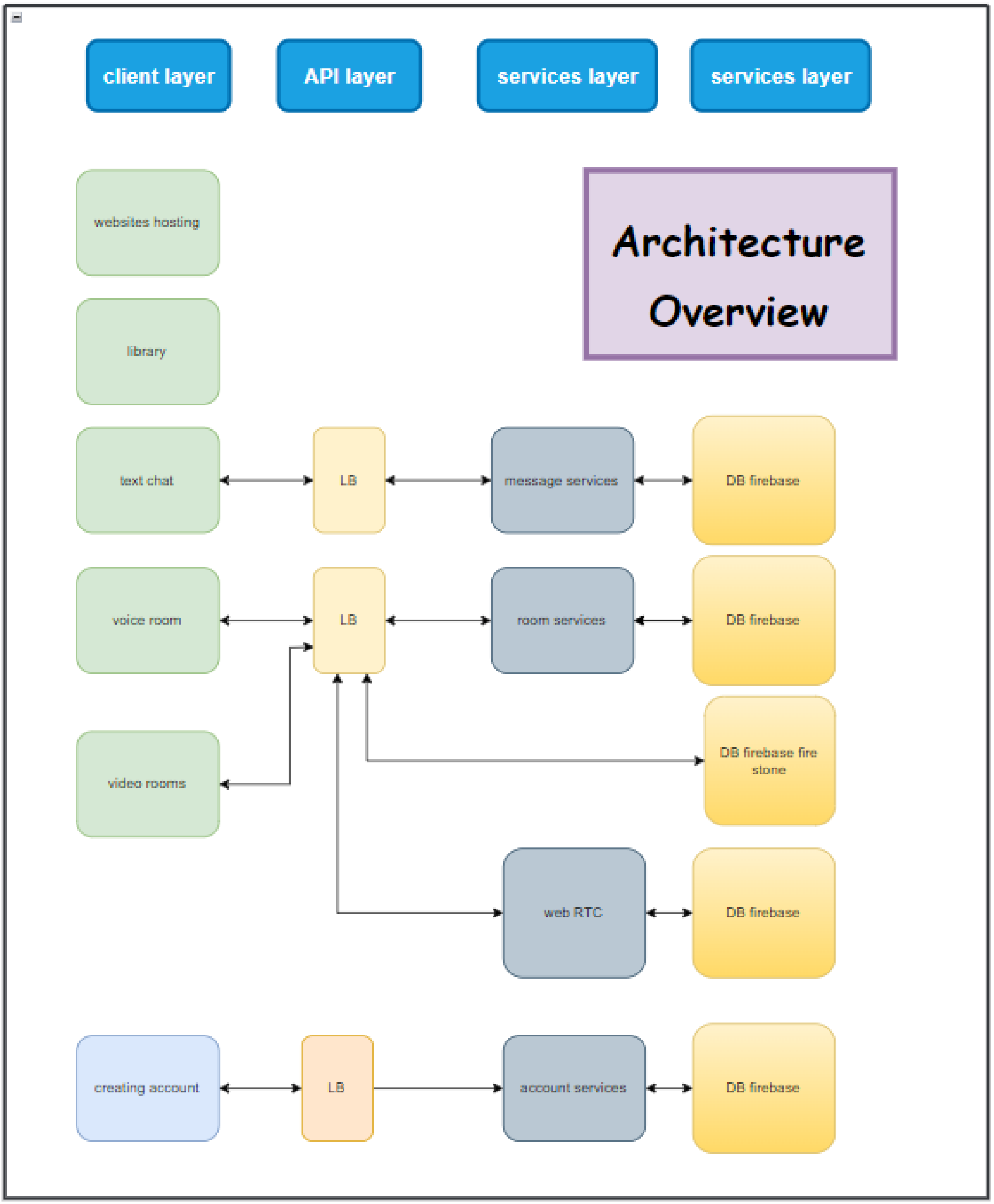
    Our website is specially for ***students***.

- ## Installation Guide:
  - ### Instructions for installation:

    1- Download the project folder and extract it.
    2- Make sure you have downloaded Vue.js .
    3- Open the *Studiando* folder using visual studio code, then open the terminal of the IDE, and write “npm install” in the terminal.
    4- Write “npm i serve” then “npm run serve” in the terminal to run the project.

- **User Manual**:
  - How-to guides or tutorials describing the flows:
    - Please use this link to go to a How-To-Use video that will answer all your questions. **https://youtu.be/sisjwQmRoyY**
    - Main page should be run on local host 8080 and user page and User page should be run on local host 8081.

- **Architecture Overview**:
    - High-level architecture diagram:
    - Components, modules, or layers and their interactions:



- ## API documentation:

Base URL: https://textchat-159f4-default-rtdb.firebaseio.com

Endpoint 1

Create new Notifications Admin sends a new notification to users
URL: “/chats/notifications.json”
Method: “Post”
Parameters:

- Index: id of the notification
- date: date of the notification
- message: content of the notification
- name: name of the sender (the admin)

Status Codes

- 201 Created: Data successfully created.
- 400 Bad Request: Invalid request parameters.
- 500 Internal Server Error: Server error.

Endpoint 2

Receive all Notifications Users receive notifications from admin
URL: “/chats/notifications.json”
Method: “Get”

Parameters:

- notifications: array of notifications

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

Endpoint 3

Create new message in a specific channel Users sends a new message in a channel by his own selection

URL: “/chats/channels/${this.currentChannel}.json”

Method: “Post”

Parameters:

- Index: id of the notification
- date: date of the notification
- message: content of the notification
- name: name of the sender (the admin)
- this.currentChannel: the selected channel

Status Codes

- 201 Created: Data successfully created.
- 400 Bad Request: Invalid request parameters.
- 500 Internal Server Error: Server error.

Endpoint 4

Receive all Channels

Users receive Channels which have been created before

URL: “/chats/channels.json”

Method: “Get”

Parameters:

- channels: array of channels

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

Endpoint 5

Create a new message in a specific private channel chat

Users sends a new message in a channel which contains 2 members only

URL: “/chats/private/${this.privateChat}.json”

Method: “Post”

Parameters:

- Index: id of the notification
- date: date of the notification
- message: content of the notification
- name: name of the sender (the admin)
- this.privateChat: the selected private chat

Status Codes

- 201 Created: Data successfully created.
- 400 Bad Request: Invalid request parameters.
- 500 Internal Server Error: Server error.

Endpoint 6

Receive all messages in a private chat

URL: “/chats/private/${this.privateChat}.json”

Method: “Get”

Parameters:

- messages: array of messages
- this.privateChat: the selected private chat

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

<u>Endpoint 7</u>

Receive all messages in a Channel
URL: “/chats/private/${this.currentChannel }.json”
Method: “Get”

Parameters:

- messages: array of messages
- this.currentChannel: the selected channel

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

<u>Endpoint 8</u>

Receive all private Chats
URL: “/chats/private.json”
Method: “Get”

Parameters:

- messages: array of all private chats

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

Endpoint 9

Receive all users in the system
URL: “/users.json”
Method: “Get”

Parameters:

- users: array of all private chats

Status Codes

- 200 OK: Successful request.
- 400 Bad Request: Invalid request parameters.
- 404 Not Found: Data not found.
- 500 Internal Server Error: Server error.

Endpoint 10

Sign in or sign up
URL: “/users.json”
Method: “Post”
Parameters:

  - fName: first name of the user
  - lName: last name of the user
  - userName: user name of the user
  - password: password
  - email: user email
  - dateOfBirth: user birth of birth
  - gender: user gender

Status Codes

- 201 Created: Data successfully created.
- 400 Bad Request: Invalid request parameters.
- 500 Internal Server Error: Server error.

Endpoint 11

Delete a user in the system
URL: “/users/ ${ourMessage}.json”
Method: “Delete”
Parameters:

- ourMessage: id of the user to be delete

Status Codes

- 204 No Content: User successfully deleted.
- 404 Not Found: User with the provided ID not found.
- 500 Internal Server Error: Server error.