ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

ΥΠΗΡΕΣΙΟΣΤΡΕΦΕΣ ΛΟΓΙΣΜΙΚΟ

- **Λειβαδάρος Παναγιώτης | Π21085**
  panoslivadaros@gmail.com
- **Παπανικολάου Νικόλαος | Π21130**
  npapanikolaou2003@gmail.com
- **Σοπιλίδης Στέφανος | Π21153**
  stefanossopilidis2003@gmail.com

7ο εξάμηνο

Table of Contents

Repositories

Lab Portal

https://github.com/NikolasPapanikolaou/final_project_

ToDo Rest API

https://github.com/StevenSopilidis/TodoAPI

Lab Portal

Το Lab Portal είναι μια Node.js εφαρμογή που παρέχει στους χρήστες μια πλατφόρμα για τη διαχείριση επαφών, τη δημιουργία αναρτήσεων, τη συμμετοχή σε ομαδικές συνομιλίες και την αποστολή μηνυμάτων. Η εφαρμογή χρησιμοποιεί το Firebase για έλεγχο ταυτότητας και λειτουργίες βάσης δεδομένων σε πραγματικό χρόνο.

Usage

1. Το API το οποίο είναι απαραίτητο για την λειτουργεία του frontend βρίσκεται στο URL: https://finallabportal.onrender.com. Εναλλακτικά σε περίπτωση που θέλουμε να το τρέξουμε τοπικά ξεκινήστε τον διακομιστή backend ():

node app.js

2. Ανοίξτε τα αρχεία HTML frontend στο πρόγραμμα περιήγησής σας (e.g., `front/html/home_page.html`, `front/html/sign_up.html`).

API Endpoints

Authentication

- **POST /signup**
  - Request body: `{ email, password, firstName, lastName }`
  - Response: `{ message, uid }`

- **POST /google_sign_in**
  - Request body: `{ uid, email, firstName, lastName }`
  - Response: `{ message }`

- **POST /login**
  - Request body: `{ email, password }`
  - Response: `{ message, token, uid, firstName, lastName }`

Contacts

- **POST /add_contact**
  - Request body: `{ currentUserUid, contactEmail }`
  - Response: `{ message }`

- **POST /accept_contact**
  - Request body: `{ currentUserUid, contactUid, requestKey }`
  - Response: `{ message }`

- **POST /get_pending_requests**
  - Request body: `{ currentUserUid }`
  - Response: `{ pendingRequests }`

- **POST /get_user_contacts**
  - Request body: `{ userId }`
  - Response: `{ contacts }`

Posts

- **POST /create_post**
  - Request body: `{ userId, title, text, contactIds }`
  - Response: `{ message }`

- **GET /get_posts**
  - Response: `{ posts }`

- **POST /get_user_posts**
  - Request body: `{ userId }`
  - Response: `{ posts }`

- **POST /get_user_feed**
  - Request body: `{ userId }`
  - Response: `{ posts }`

- **DELETE /delete_post**
  - Request body: `{ postId }`
  - Response: `{ message }`

Group Chats

- **POST /create_group_chat**
  - Request body: `{ userId, groupName, memberIds }`
  - Response: `{ message }`

- **POST /get_user_group_chats**
  - Request body: `{ userId }`
  - Response: `{ groupChats }`

- **POST /send_message**
  - Request body: `{ chatId, userId, message }`
  - Response: `{ message }`

- **POST /get_messages**
  - Request body: `{ chatId }`
  - Response: `{ messages }`

Socket Events

- **joinRoom**
  - Payload: `{ userId }`

- Description: Joins a room with the user's ID.

- **newMessage**
  - Payload: `{ chatId, userId, message }`
  - Description: Sends a new message to the specified chat.

- **message**
  - Payload: `{ chatId, groupName, email, text }`
  - Description: Receives a new message in the chat.

Lab Portal Frontend

Αυτό το έγγραφο παρέχει μια επισκόπηση και εγχειρίδιο χρήσης για το frontend του έργου Lab Portal. Το frontend είναι κατασκευασμένο χρησιμοποιώντας HTML, CSS και JavaScript και αλληλεπιδρά με έναν διακομιστή backend για να παρέχει λειτουργίες όπως εγγραφή χρήστη, σύνδεση και συνομιλία.

Usage

Sign Up

1. Ανοίξτε το αρχείο 'sign_up.html' στο πρόγραμμα περιήγησής σας.
2. Συμπληρώστε τα απαιτούμενα πεδία: Όνομα, Επώνυμο, Email και Κωδικός πρόσβασης.
3. Κάντε κλικ στο κουμπί "Εγγραφή" για να δημιουργήσετε έναν νέο λογαριασμό.
4. Εάν η εγγραφή είναι επιτυχής, θα μεταφερθείτε στη σελίδα σύνδεσης.

Log In

1. Ανοίξτε το αρχείο 'log_in.html' στο πρόγραμμα περιήγησής σας.
2. Συμπληρώστε το email και τον κωδικό πρόσβασής σας.
3. Κάντε κλικ στο κουμπί "Σύνδεση" για να συνδεθείτε στο λογαριασμό σας.

4. Εναλλακτικά, μπορείτε να συνδεθείτε χρησιμοποιώντας τον λογαριασμό σας Google κάνοντας κλικ στο κουμπί "Σύνδεση με Google".

5. Εάν η σύνδεση είναι επιτυχής, θα μεταφερθείτε στην αρχική σελίδα.

Home Page

1. Αφού συνδεθείτε, θα μεταφερθείτε στο 'home_page.html'.

2. Η αρχική σελίδα εμφανίζει τα αιτήματα φιλίας σας, τα εκκρεμή αιτήματα, τις αναρτήσεις σας και τη ροή σας.

3. Μπορείτε να στείλετε ένα αίτημα φιλίας εισάγοντας ένα email και κάνοντας κλικ στο κουμπί "Αίτημα".

4. Μπορείτε να δημιουργήσετε μια νέα ανάρτηση κάνοντας κλικ στο κουμπί "+", συμπληρώνοντας τα στοιχεία της ανάρτησης και κάνοντας κλικ στο "Δημιουργία ανάρτησης".

5. Μπορείτε να κάνετε αναζήτηση στη ροή σας χρησιμοποιώντας τη γραμμή αναζήτησης στο επάνω μέρος της ενότητας ροής.

Conversation Page

1. Για πρόσβαση στη σελίδα συνομιλίας, κάντε κλικ στο κουμπί συνομιλίας (εικονίδιο φυσαλίδας ομιλίας) στην αρχική σελίδα.

2. Η σελίδα συνομιλίας σάς επιτρέπει να δημιουργείτε ομαδικές συνομιλίες και να στέλνετε μηνύματα.

3. Μπορείτε να δημιουργήσετε μια νέα ομαδική συνομιλία συμπληρώνοντας το όνομα της συνομιλίας, επιλέγοντας μέλη και κάνοντας κλικ στο "Δημιουργία συνομιλίας".

4. Μπορείτε να στείλετε μηνύματα στην επιλεγμένη συνομιλία πληκτρολογώντας την εισαγωγή μηνύματος και κάνοντας κλικ στο "Αποστολή".

File Structure

- `front/html/`
  - `sign_up.html`: Sign-up page.
  - `log_in.html`: Log-in page.
  - `home_page.html`: Home page.
  - `conversation.html`: Conversation page.
- `front/js/`

- `sign_up.js`: JavaScript for the sign-up page.

- `log_in.js`: JavaScript for the log-in page.

- `home_page.js`: JavaScript for the home page.

- `conversation.js`: JavaScript for the conversation page.

Notes

- Βεβαιωθείτε ότι ο διακομιστής backend εκτελείται και είναι προσβάσιμος στο καθορισμένο 'API_URL' στα αρχεία JavaScript.

- Το frontend χρησιμοποιεί το Firebase για έλεγχο ταυτότητας Google. Βεβαιωθείτε ότι η διαμόρφωση του Firebase έχει ρυθμιστεί σωστά στο "log_in.js".

Δημιουργία REST API

Παρακάτω, παραθέτουμε ορισμένες σημειώσεις οι οποίες αφορούν την λειτουργία της εφαρμογής του 2ου θέματος, οι οποίες είναι απαραίτητες για την εκτέλεσή της.

Αρχικά, δεν έχουμε υλοποιήσει API για τη διαδικασία Logout του χρήστη, καθώς η διαδικασία αυθεντικοποίησής του γίνεται μέσω jwt bearer tokens. Συνεπώς, ο ίδιος ο χρήστης είναι υπεύθυνος για τη διαγραφή του παραπάνω token.

Ακόμα, αναφέρουμε πως το endpoint για τη διαδικασία Login του χρήστη, είναι τοποθετημένο στην παρακάτω διαδρομή: /user/signup.

Επιπρόσθετα, η εφαρμογή «ακούει» στη θύρα 8080 του υπολογιστή, ενώ πρέπει να έχουμε δημιουργήσει και μία Βάση Δεδομένων σε PostgreSQL, με τα εξής παρακάτω credentials:

Host=localhost;Port=5432;Database=TodoAPIDB2;Username=root;Password=test;

Τέλος, μπορούμε να έχουμε πρόσβαση στο ζητούμενο swagger url, χρησιμοποιώντας το παρακάτω url όσο τρέχει η εφαρμογή: http://localhost:8080/swagger