Prepared for: She codes Final project

Prepared by: Nataly Basovich

30 June 2022

Doc number: ver 1.0

Design doc

Introduction

Scope

The application will help traveler to prepare, plan and document the trip without forgetting important things, and be ready during the whole trip and in its each step.

Overview

My project will consist of front end, backend and db.

- Front end will be built with React, Redux and Material UI design system.

- BackEnd: Node JS with Express.

- DataBase: MongoDB, will contain trips information for each user and user data for authentication.

 In front end will be described different screens of the application.

Terminology

Trip plan means that user is planning his travel date, stay places, transports and places to visit. So that the user can get the best price for his early ordering of the plane and the hotel, for example.

- Preparation means that the user prepares the list things that he needs to take with for this trip.

- Transportation. Here user selects which types of transport he prefers to get to the place of trip. Also user can add transport that he will need into the city, for example, any card for underground and etc.

- Stays. The user creates a list of places where he wants for stay in and selects the most suitable before the trip.

- Visiting. Here the user creates list places that he wants to visit ordered by date and with option to add notes.

- Expenses. Here the user documents all the expenses during his trip in different currencies and different payment methods. He can get the summary of his expenses by currencies and payment methods in the view mode of this city.

Software design description

Generel flow

The user can create multiple trips, view, edit or delete them. Also user can add or delete different options for each trip and their explaination will be given below.

Software architecture

UI architecture

Description components.

The application has the following screens.

- **Initial screen**
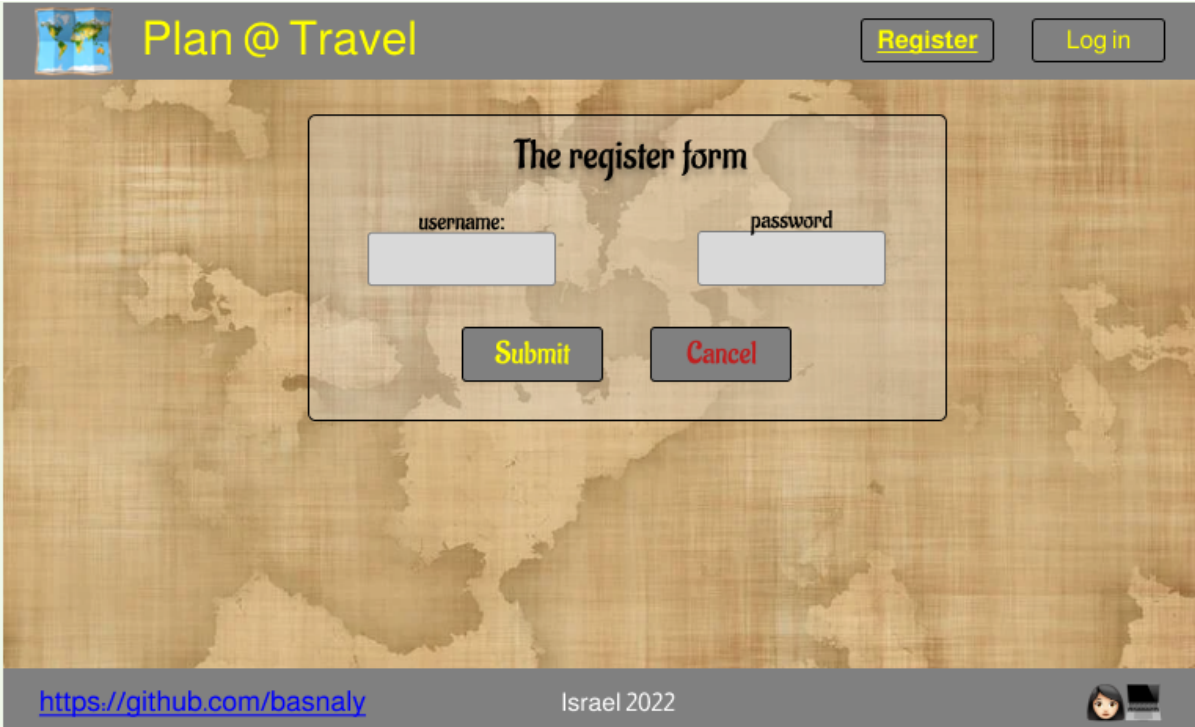
There is an input of city - text (string), two inputs of period - dates.

After adding this, user have to register or login to continue with the planing.

The route is: /home.
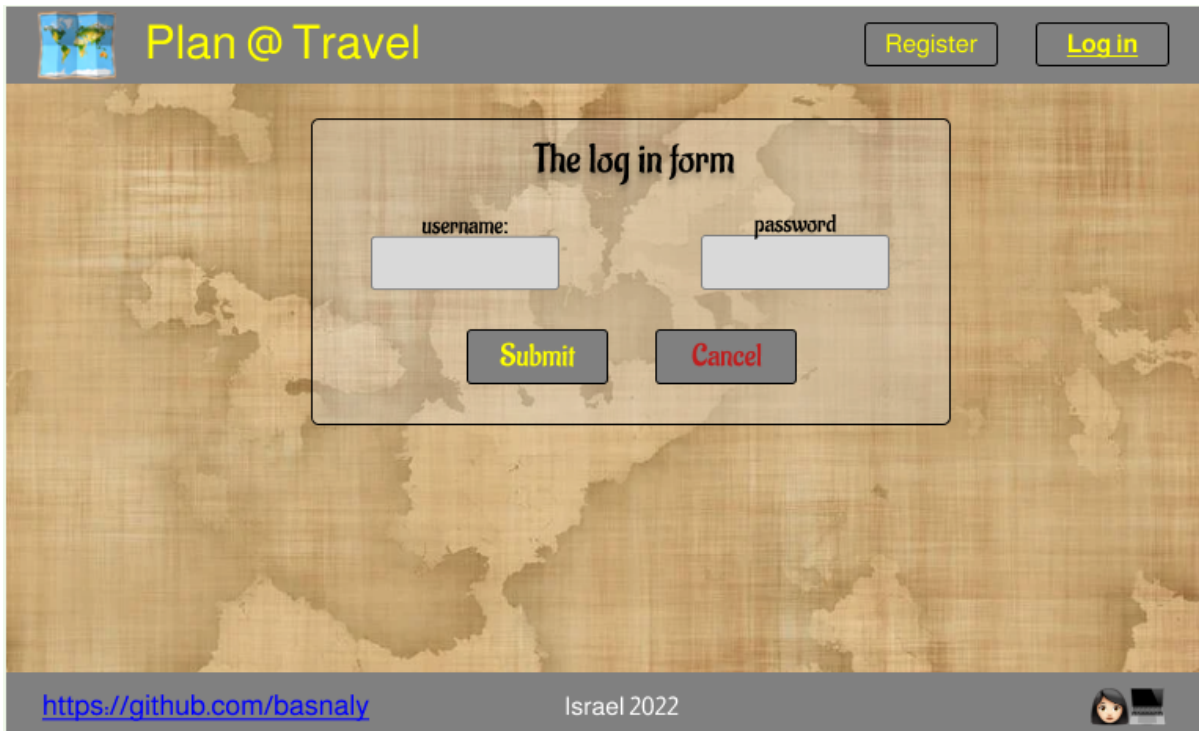
• **Register screen:**

Input of username - email.

Input of password - text, it must includes letters with upper and lower cases, integers and symbols.

After finishing the registration, the user email and password will be added to the db table of users in Mongo DB.

The route is: /register.

• **Log in screen:**

Input of username - email.

Input of password - text (string), it must include letters with upper case and lower case, integers and symbols.

The route is: /login.

- **Main screen for new user:**

After the user gets registered, he moves to the edit trip screen, where he can fully update the trip he created in the initial screen.

The city and the date of travel is shown in the main screen according to the initial screen.

The user can change the dates and save the all the changes using save button located on the upper part of screen.

The route is: /London_id/period.

- **Main screen - preparations:**

In this screen user can create categories like 'Documents', 'Clothes', 'Care products' end etc. Also in each category the user can add notes - check items. User also can add and remove notes and categories.

The route is: /London_id/preparations.

Every tab is pretty similar to the preparations tab, it will allow user to add an essential information. If necessary I'll add the remaining screens before lesson at 04/07.

- **Main screen for existing user:**

This screen allows user to select city from the list and interact with them - edit, view or delete.

- View. After user clicks on the view button the route changes to /city_id/view and user can see the full info about his trip. It allows to see all the data in one screen.

- Edit. After user clicks on the edit button the route changes to /city_id/period. Also user can edit data on each of the nav bar tabs.

- Delete. After user clicks on the delete button, he gets modal if he really wants to delete all the city data. And if yes the city is getting deleted.

Backend architecture

Back end routing are:

- **post('/register') - validate and save user data in Mongo db.**
'/register' gets username and password and returns error if the password is not valid or cannot be saved in the data base. If there are no errors returns token for the next interactions with the server.

- **post('/login') - authenticate user.**
'/login' gets username and password, compare if the username and the password are much to existing user create a new token and return it. If not returns an error.

In every interaction with the server, the server validates the token.

- **get('/list_cities') - get data of cities of this user.**
'/list_cities' returns error if cannot get the data or the list cities from the data base.

- **post or delete('/delete_city') - delete city from Mongo db.**
'/delete_city' gets city_id and returns success if deleted or error if not.

- **post('/city_id_edit') - edit trip.**
'/city_id_edit' gets all the data of city_id and returns if update of the data in the data base was successful or error if not.

First project steps:

- **Next week (05 - 12/07):**
1. Build authentication process in Google Firebase.
2. Build Navbar with tabs: period, preparations, transportations, stays visitings and expences using Material UI.

- **Next week (12 - 19/07):**
1. Built mock trip with data for 2 tabs in UI.
2. Use Formik library to edit data for tabs above.