**טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**Virtual Event Planning Platform**

**Group A2**

|  |  |
| --- | --- |
| **חבר.ת צוות** | **ת.ז** |
| Or Balmas | 205907926 |
| Semion Rudman | 319636395 |
| Anastasya Chesnov | 317450013 |
| Marina Shteinfer | 323305458 |

**<תקציר הפרויקט - עד חצי עמוד- פונקציונליות מרכזיות לפי משתמשים>**

A platform for organizing and managing virtual events. Implement functionalities like event scheduling, participant management, and real-time communication tools.

**Important Code parts:**

The code is consist of 3 main parts:

1. **Login:**

* const login
* Input: username, password
* Output: next page or error message
* Process: Takes Credentials and checks if they exist in the “Users” scheme in the DB. If they are exits, the user gets to the next page, if not, an error will appear.
* const Registration:
* Input: Username, password and confirm password
* Output: Error message if failed to register, and success message if succeeds.
* Process: Takes input from the user and checks if the password and the confirmed password are the same, if they are, saves information in the “Users” scheme, if not, sends an error message to the user.

1. **Main event management page:**

* const addEvent:
* Input: Event name, date, time
* Output: If succeeds, sends a success message, if fails, sends an error message
* Process: Takes the input and saves it in the “events” scheme in the DB. The “username” part the function takes not from user input but checks which user created the event. It also updates the calendar view. If the creation fails, the function shows the user an error message.
* const generateCalendarDays:
* Input: Year, month
* Output: A calendar in the current month and year with all the events that the user has created or participate according to colors.
* Process: Takes the “year”, “month” and “username”, and check with the “event” scheme if there are events that the user created. Next checks for each event that exists if the user attends them, and displays these events on the calendar.
* const EventPage:
* Input: User and dates
* Output: The event page structure
* Process: Takes the input, and base on it fetches the data from “users”, to display the username connected. Also based on user information display the events that connect to the user or the user can join.
* const JoinEvent:
* Input: Event name, username
* Output: If fails an error message, if succeeds a success message.
* Process: Takes username and the event name of the event the user wants to join. Checks if the event exists in the “events” scheme in the DB. if exist add user to the attendees array.
* const sendMessage:
* Input: String message from user, event name, username and timestamp
* Output: The message in the dialogue box with messages from other users in the event. If there is an error, it displays an error message.
* Process: Takes the input and saves it in the “messages” scheme in DB. also fetches messages for the same event and displays them.

1. **Event editor page:**

* const EditEvent:
* Input: Changes from the user, in event name, time, date, location, description and attendees.
* Output: Event with the updated changes in the main page, or error message if the process fails.
* Process: Fetches the data of the selected event from “events” scheme and displays it to the user. The user enters his changes and the application updates the DB. If the user deletes the event, it also removes it from the “events” scheme.

**< מימוש-שמות הטכנולוגיות המרכזיות בכל אחד מהחלקים -/styling/db/ Backend/frontend>**

**Visual studio code:**

For this web project vs code was used as a development environment. Vs code is a streamlined code editor developed by microsoft for windows, linux and macOS. It combines dubbaging, syntax highlighting, intelligent code completion, and snippets code refactoring features. Vs code can be used with variety of programming languages including: Java script, html, css, java, c, c# and ect. In addition it also supports debugging in Nodejs. With Html the Vs code can show how the application works on browsers with “live server” service. It can also work also with GitHub.

**MongoDB:**

MongoDB is a cross-platform and document-oriented database program. It is classified as NoSql DB and utilizes JSON files with the option of scheme. MongoDB is used to store information for our application.

**Our database consists of three JSON files:**

***Users***: consist of the following information:

1. username
2. Password
3. Id ->key

***Events***:

1. name
2. Id -> key
3. date
4. time
5. Username - event creator
6. Attendees - array

***messages***:

1. Id -> key
2. eventName
3. Sender
4. Text
5. Timestamp

**Vercel**:

Used in this project as a deployment environment. Vercel is a cloud platform as service. It maintains the Next.js web development framework. In vercel deployments are handled through Git repositories.

**Tailwind Css:**

Used in this project to design the website. Tailwind is a Css framework that allows design with predefined classes. Tailwind is a utility-first CSS framework which means we can use utility classes to build custom designs without writing CSS as in the traditional approach. It also allows extensive customization and avoids pre-built component styles, offering flexibility in design and eliminating the need for writing custom CSS, resulting in smaller file sizes and faster load times.

Tailwind can make in a simple way responsive design with built-in classes, helping the creation of mobile-friendly layouts.

<https://github.com/MarinaShtei/A2>

<https://a2-alpha-ashen.vercel.app/>

<<https://www.morethanwallet.com/app/342>>

**Functional requirements:**

1. Users will be able to schedule virtual events by specifying the date, time, and type of event.

2. Organizers will manage attendee lists.

3. Platform will include user-to-user messaging.

4. Platform will include a user registration system.

5. Platform will be able to categorize events based on type, topic, or other criteria, and provide users with filtering options.

6. Organizers will be able to edit existing events that they created.

7. Users will be able to join events.

**Non-functional requirements:**

1. Usability: The platform should have a user-friendly interface that is easy to navigate and understand.

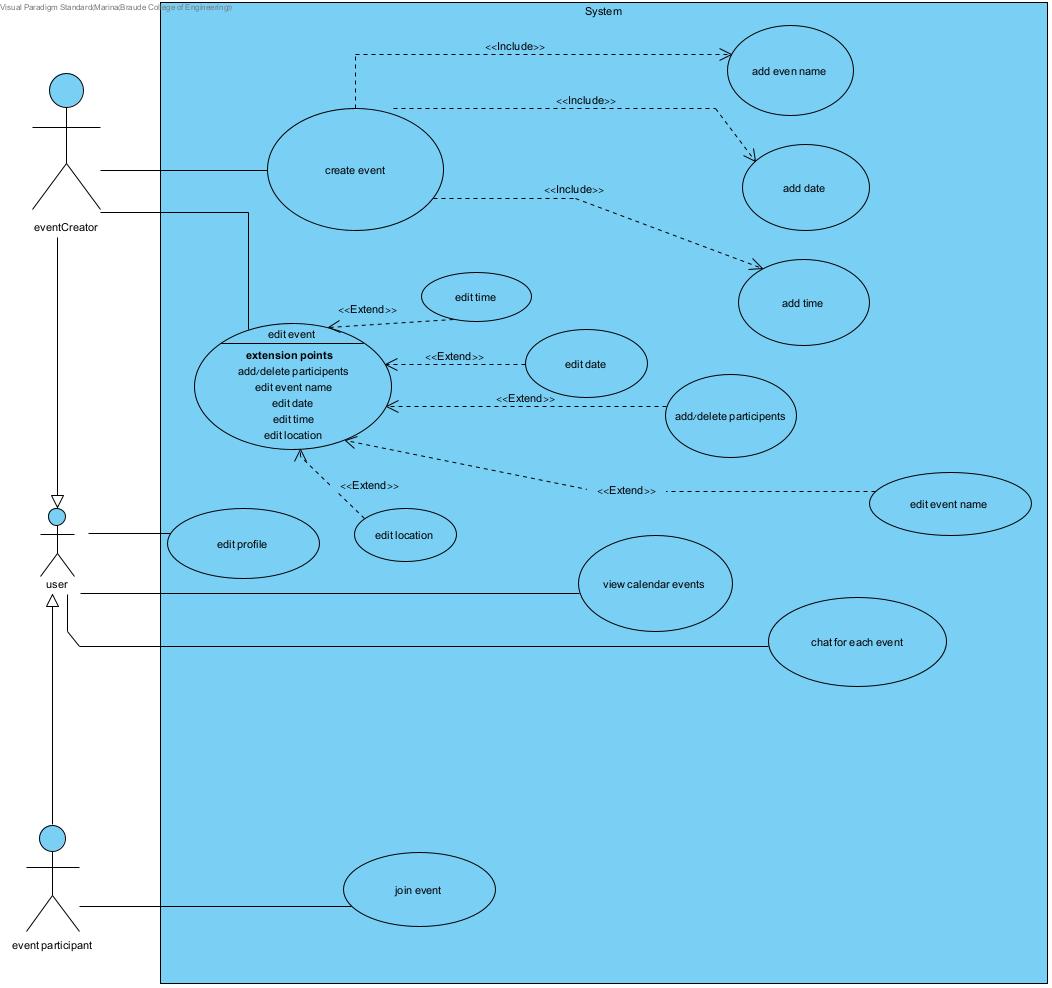
2. Security: Implement user login functionality to prevent unauthorized access to data.

3. Reliability: The platform should be available whenever users need to access it.

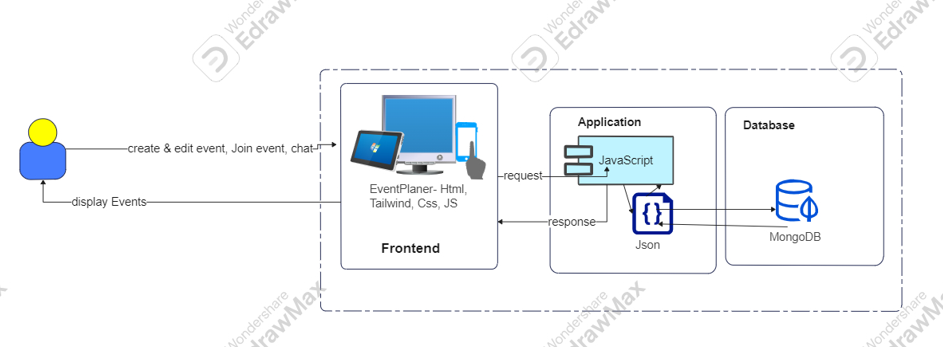
4. Compatibility: The platform should be compatible with a range of devices and web browsers to ensure accessibility for users.

5. Documentation: The platform should include comprehensive documentation for both users and developers, including user guides.

**Use case:**



**Architecture of the site:**

****

**User guide:**

https://docs.google.com/document/d/1F37NAeWXaT9dsULjBfMVgvqs9uphf-CpBNMGMudZoXM/edit

**Maintenance guide:**

https://docs.google.com/document/d/1q8EhKBIj1fmBy82quLLtqDEZVSXFhgOnzed-BnNb\_x4/edit