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

**הגשת פרויקט**

**TicTacTask** 12 **A**

|  |  |
| --- | --- |
| שם חבר.ת הצוות | תז |
| אילנית חנוכו | 208657320 |
| נוי ביטון | 312238678 |
| מעיין שרביט | 318301744 |
| חן חנצין | 208636761 |

**תקציר הפרויקט:**

TicTacTask is a management tool that designed to streamline team productivity by enabling task management, progress monitoring, and real-time document collaboration. It combines an intuitive interface with tailored functionality.

1. Task Assignment:  
- Set priorities to each task.

- Set deadlines to each task.  
- Set category to each task.

2. Progress Tracking:  
- Use dashboards to visualize real-time progress with progress bar.

- Track and update task statuses.

3. Collaborative Document Editing:  
- Edit documents simultaneously and see changes in real time.

4. Integrations:  
- Integrate with tools like Google Drive.  
  
The tool aims to enhance project management efficiency and foster effective team collaboration through its robust and user-friendly functionalities.

**מימוש- הטכנולוגיות המרכזיות בפרויקט:**

Styling – tailwind  
DB – MongoDB Atlas

Backend – javascript, nodejs, express

Frontend - javascript, React, html

**קישור לתיקיית גיט ציבורי:** <https://github.com/ilanithanooko/A12.git>

**קישור לאתר:** <https://tictactask-app.onrender.com>

**קישור ל -:MTW** <https://www.morethanwallet.com/app/303>

**שם משתמש וסיסמה לדוגמה לכניסה לאתר:**

Email: maayansharvit1997@gmail.com  
Password: KatyPerry2il@1

|  |  |  |
| --- | --- | --- |
| Team Member | Designated Tasks | Finished Tasks |
| Noy+Maayan | Preparation of a use case diagram describing the use of the website. | Done |
| Ilanit+Hen | A presentation of the website's architecture | Done |
| Ilanit | Homepage functionality after the login  Preparation of a use case diagram describing the use of the website. | Done |
| Hen | User Authentication Implementation: Develop and integrate user authentication functionality to ensure secure access to the project management tool. | Done |
| Hen | Login page | Done |
| Noy+Maayan | Database setup | Done |
| Noy + Ilanit | Frontend-backend integration | Done |
| Hen | Collaborative document editing feature development | Done |
| Maayan | Setting up and overseeing testing | Done |

1. Our system engineer would be Noy:

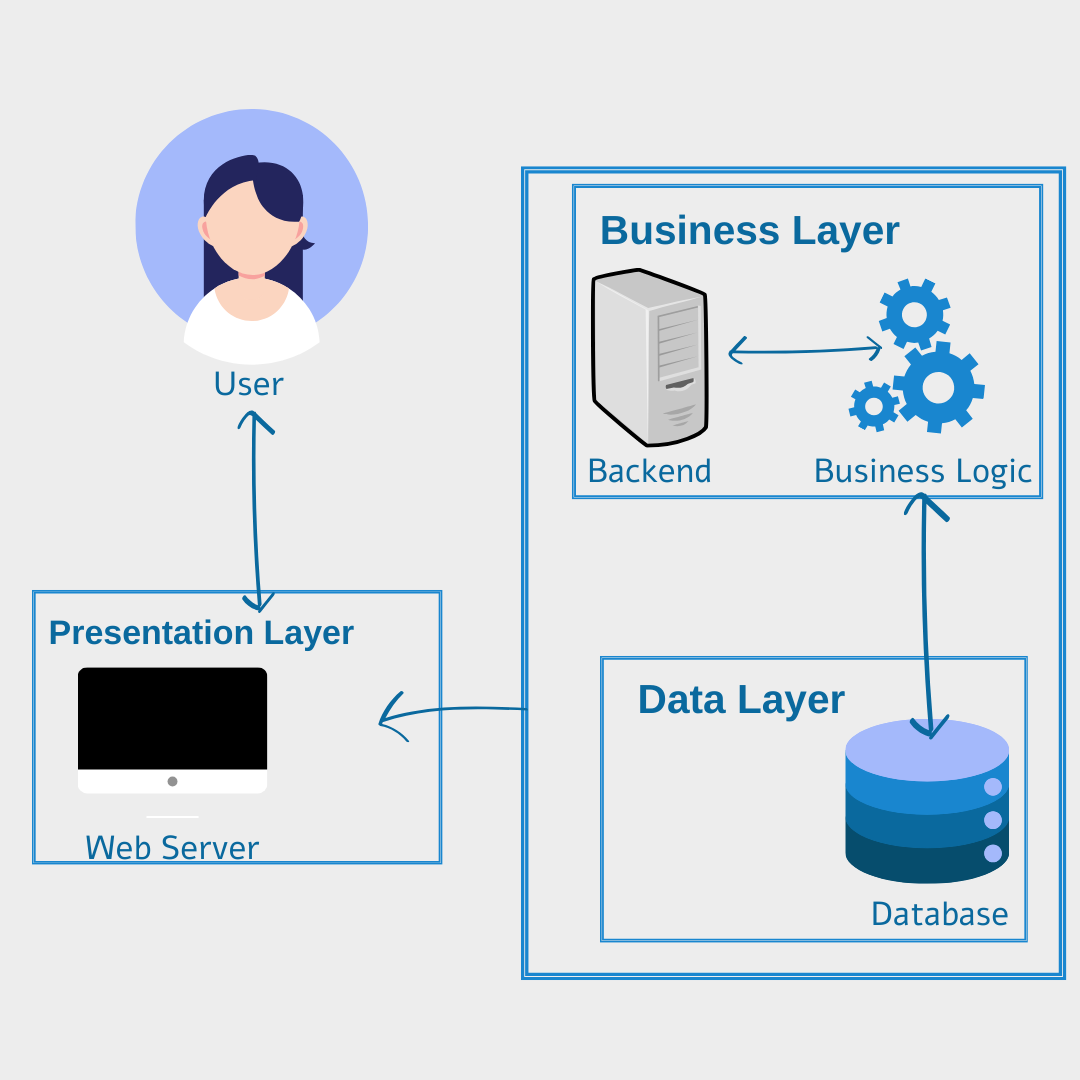
2. Functional and Non-Functional Requirements

**Functional Requirements:**

1. The Application shall provide user login.
2. The Application allows the user to add a new task.
3. The Application allows the user to edit an existing task.
4. The Application allows the user to delete an existing task.
5. The Application displays task progress.

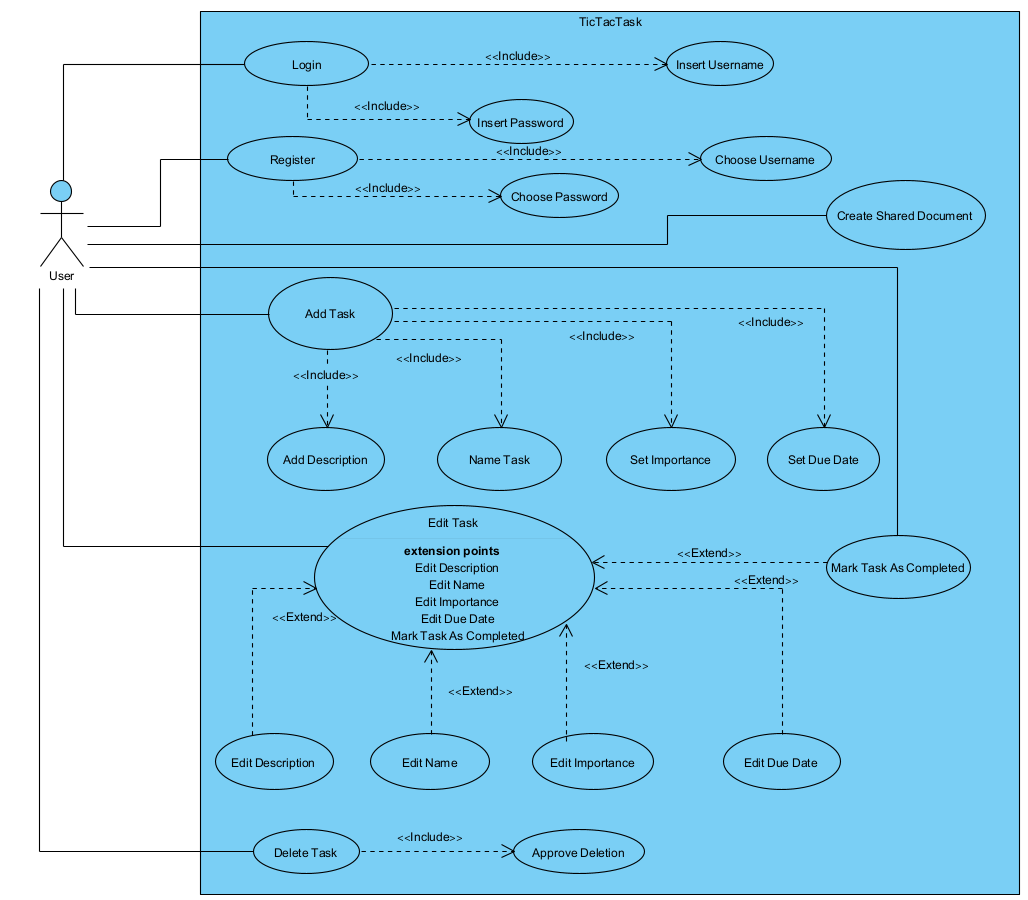
**Non-Functional Requirements:**

1. Usability-The user can add a new task by clicking on the ‘plus’ button.
2. Usability-The user can edit an existing task by clicking on the ‘edit’ button.
3. Usability-The user can delete an existing task by clicking on the ‘delete’ button.
4. Privacy-The user can login into the application by entering username and password.
5. Usability-Progress rate can be seen by the number of completed tasks out of the total amount of tasks.
6. Portability- The application will be adapted to multiple platforms.
7. ארכיטקטורה

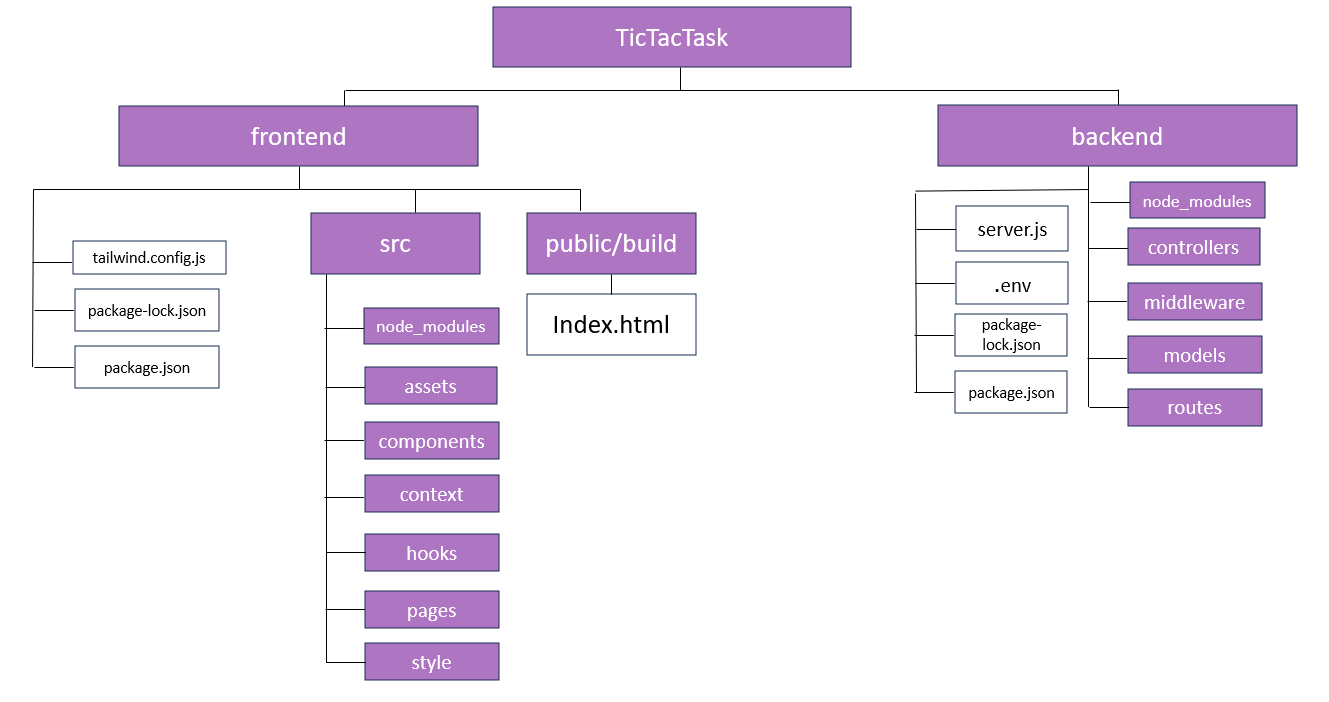
  
Presentation Layer: This is the layer with which users directly interact. It consists of a web server and the user interface (UI). The UI is rendered in the user's browser and includes all the client-side elements they can interact with.

Business Layer: This layer contains the backend server and the business logic of the application. It's responsible for processing user requests from the presentation layer, executing the appropriate business rules, and handling application operations. This is where the core functionality of the app is implemented.

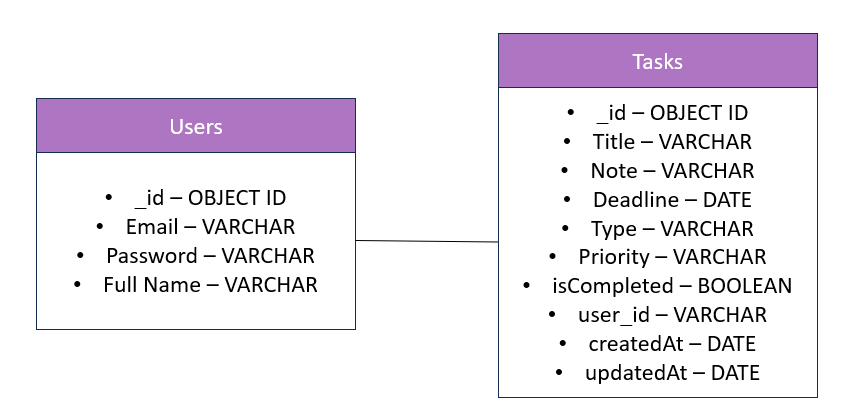
Data Layer: This is the MongoDB database where all the application data is stored. The business layer communicates with the data layer to retrieve, update, and store data based on the business logic's operations.

3. Use-Case:

4. א.



ב.



5. תיק המתכנת:

**User Authentication Endpoints:  
1. Sign-up Endpoint:**

* Path: ' /api/signup'
* Method: 'POST'
* Purpose: Allows users to create a new account by providing their first name, last name, email, and password.
* Request Body:
* 'firstName': The user's first name
* 'lastName': The user's last name
* 'email': the user's email address
* 'password': The user's chosen password
* Response:
* '201 Created': Returns a success message indicating that the user account has been created successfully.
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

**2. Login Endpoints:**

* Path: ' /api/login
* Method: 'POST'
* Purpose: Allows users to log in to their account by providing their email and password.
* Request Body:
* 'email': the user's email address
* 'password': The user's password
* Response:
* '200 OK': Returns a JSON web token (JWT) if the login is successful. This token can be used for authentication in subsequent requests.
* '401 Unauthorized': Indicates that the provided email or password is incorrect.
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

**Task Management Endpoints:  
1.Create Task Endpoint:**

* Path: ' /api/tasks
* Method: 'POST'
* Purpose: Allows authenticated users to create a new task with a title, description, deadline, priority, and type.
* Request Body:
* 'title': The title of the task
* 'description': The description of the task
* 'deadline': The deadline for completing the task.
* 'priority': The priority level of the task (low, medium, high).
* 'type': The type or category of the task (personal, work, home, educational).
* Response:
* '201 Created': Returns the created task if successful.
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

**2.Fetch Tasks Endpoint:**

* Path: ' /api/tasks
* Method: 'GET'
* Purpose: Allows authenticated users to retrieve all tasks associated with their account.
* Response:
* '200 OK' : Returns an array of tasks if successful
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

**3.Update Task Endpoint:**

* Path: ' /api/tasks/:taskId'
* Method: 'PUT'
* Purpose: Allows authenticated users to update an existing task identified by its ID.
* Request Body:
* 'title': The title of the task
* 'description': The description of the task
* 'deadline': The deadline for completing the task.
* 'priority': The priority level of the task (low, medium, high).
* 'type': The type or category of the task (personal, work, home, educational).
* Response:
* '200 OK' : Returns an array of tasks if successful
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

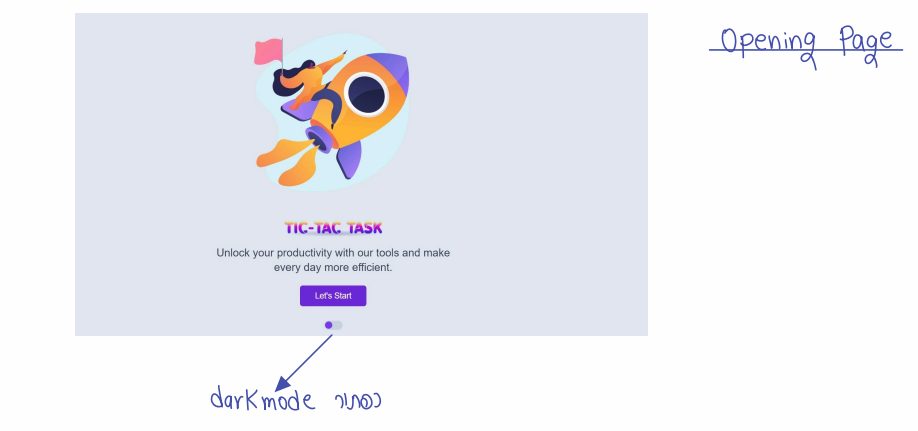
**4.Delete Task Endpoint:**

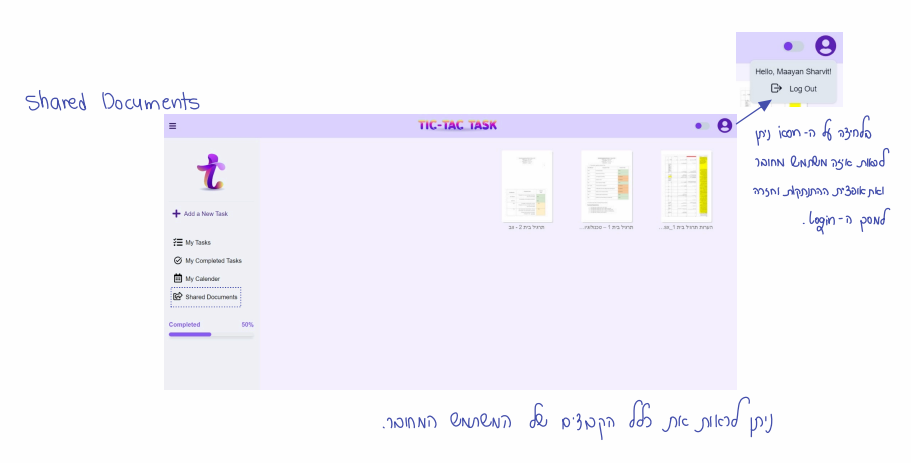
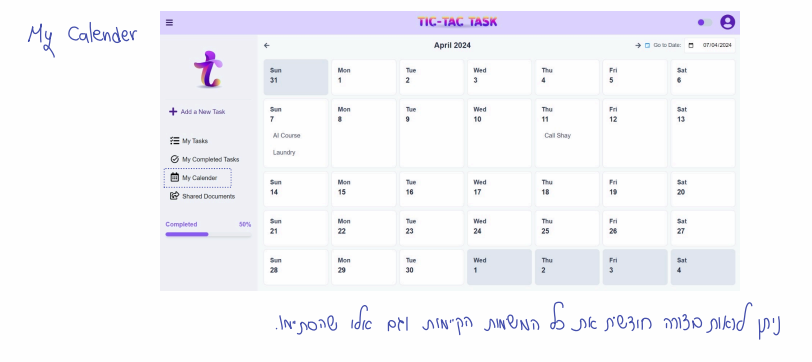
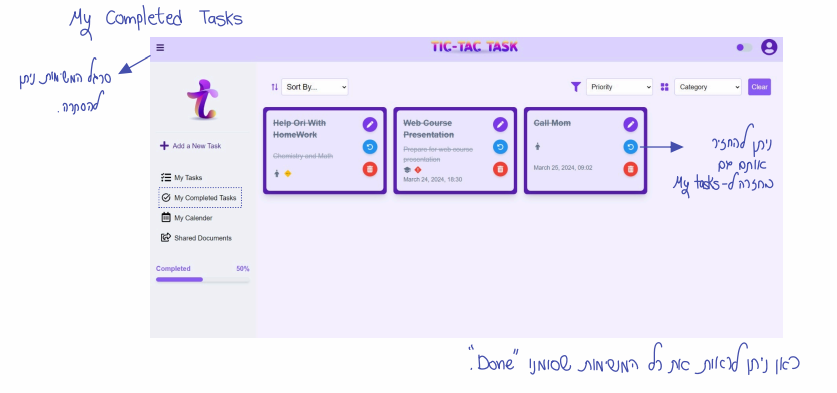
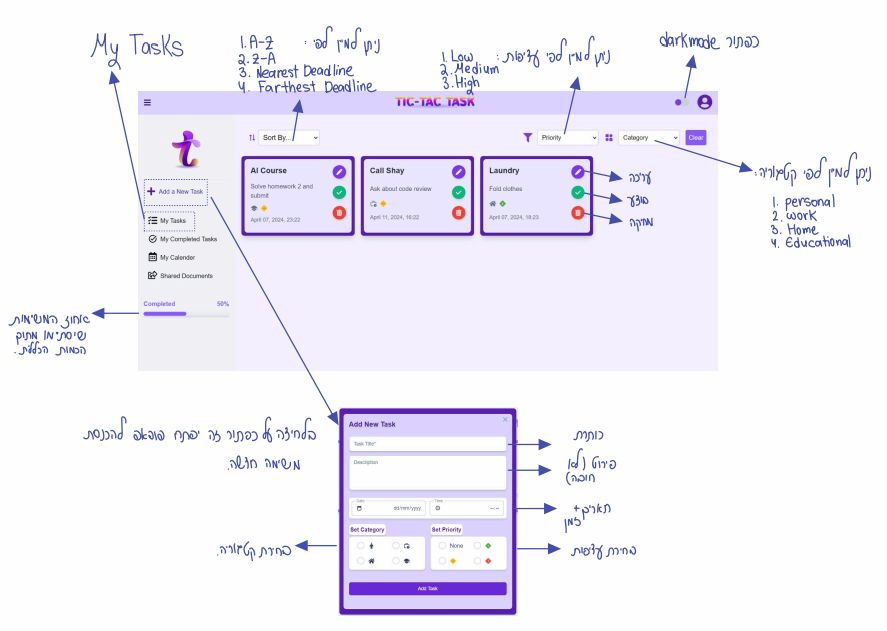
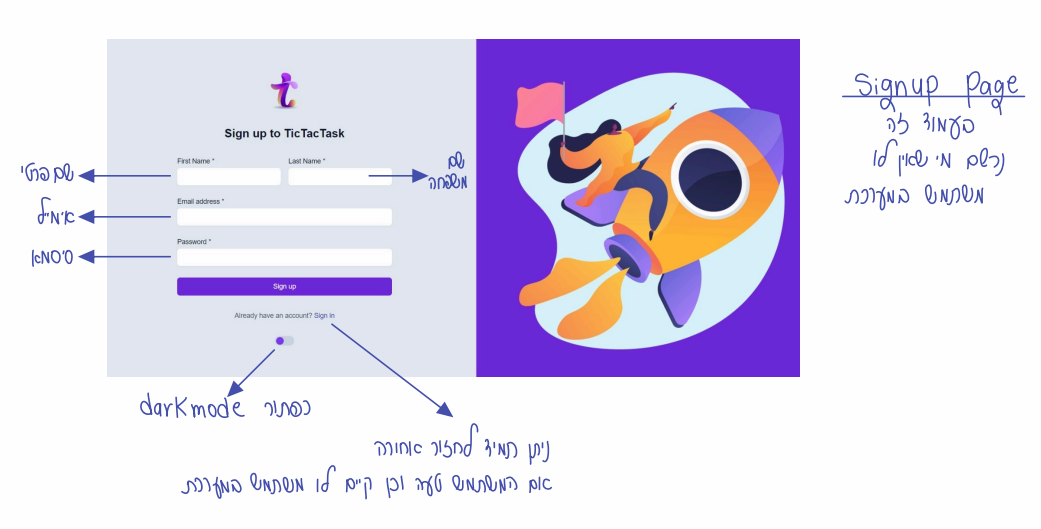
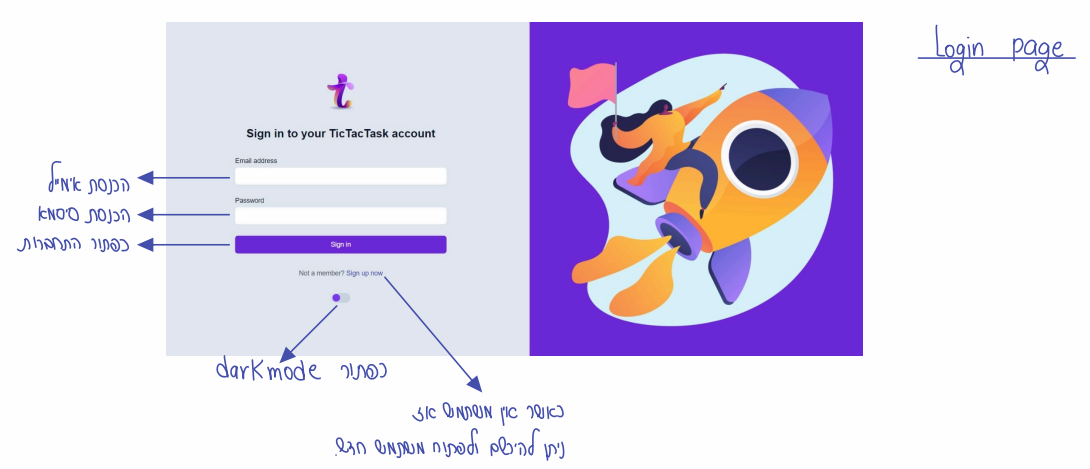
* Path: ' /api/tasks/:taskId'
* Method: 'DELETE'
* Purpose: Allows authenticated users to delete an existing task identified by its ID.
* Response:
* '200 OK' : Returns an array of tasks if successful
* '500 Internal Server Error': Indicates that an unexpected error occurred on the server.

**Development Environment:**

* Visual Studio Code: was selected because it has lots of useful add-ons, comes with a built-in terminal, and makes it easy to manage our code changes. It's great for working on both the front and back ends of our project.
* Express.js: We use Express.js for the backend of our project. It helps us handle API requests, manage business rules, and interact with the database in an organized way. We structure it using routers, business logic, and data access layers for cleaner and more efficient code.

6. תיק משתמש:





**קישור לקוד הפרויקט:**

<https://github.com/ilanithanooko/A12.git>  
  
**קישור לאתר:**  
<https://tictactask-app.onrender.com>

**שם משתמש וסיסמה לדוגמה לכניסה לאתר:**

Email: maayansharvit1997@gmail.com  
Password: KatyPerry2il@1