

Functional Requirements

1. The website must provide login services before allowing users to access Todos.

- 1.1. The website should allow an existing user to login.
 - 1.1.1. The login should take in a user's email.
 - 1.1.2. The login should take in a user's password.
 - 1.1.3. The email and password should be verified against the website's database that such a user exists.
 - 1.1.4. If the user exists and the password is correct, the user is logged in.
 - 1.1.4.1. If no such user exists, the website prompts the user to register an account.
 - 1.1.4.2. If the password is incorrect, the website informs the user that the password is incorrect.
- 1.2. The website should allow new users to register an account.
 - 1.2.1. The registration should take in a username.
 - 1.2.2. The registration should take in an email account.
 - 1.2.3. The registration should take in a password.
 - 1.2.4. The registration should take in a date of birth.
 - 1.2.5. The website should create a new user account, with the registration values given, in the database.
- 1.3. The website should allow users to change their password.
 - 1.3.1. The user must enter their email account.
 - 1.3.2. If the email account is valid, the website must send a link, to change the user's password, to that email account.
 - 1.3.3. Upon clicking the change password link, the website must prompt the user for a new password.
 - 1.3.4. Upon submitting the new password, the website must replace the user's old password in the database with the new password.

2. The website must be able to provide location services.

- 2.1. The location services must be able to track the user's live location.
- 2.2. The location services must be able to track the user's destination listed in a Todo.
- 2.3. The location services must be able to suggest a route from one location to another.
 - 2.3.1. The user must be able to view the suggested route.

3. The website must allow the user to create Todos.

- 3.1. The website must save created Todos as non-volatile data in a database.
 - 3.1.1. The Todo must take in a title from the user.
 - 3.1.2. The Todo must take in a task to be done from the user.
 - 3.1.3. The Todo must take in the datetime of the task from the user.
 - 3.1.4. The Todo must take in the duration of the task from the user.
 - 3.1.5. The Todo must take in a destination of the task from the user.
 - 3.1.5.1. The Todo must take in an indoors/outdoors value.
 - 3.1.5.2. The website must suggest the most efficient travel routes to the user, based on the destination of the Todo.

- 3.1.6. If another Todo's datetime overlaps with the new Todo, the website displays a warning, prompting the user to make changes to the current Todo's datetime accordingly.
- 3.2. The user must be able to view created Todos.
 - 3.2.1. The viewed Todo must show its title.
 - 3.2.2. The viewed Todo must show the user's task to be done.
 - 3.2.2.1. The viewed Todo must show the completion status of the task as a boolean.
 - 3.2.3. The viewed Todo must show the task's destination.
 - 3.2.3.1. The viewed Todo must show the task's indoors/outdoors value.
 - 3.2.3.2. The viewed Todo must show the route suggested by the website.
 - 3.2.4. Users must be able to select the color of Todo to indicate order of priority (e.g dark colors for urgent, light colors for non-urgent).
 - 3.2.4.1. The viewed Todo must show its priority color.
- 3.3. The user must have the option to edit the values of a created Todo.
 - 3.3.1. The Todo must be able to take in a new replacement title from the user.
 - 3.3.2. The Todo must be able to take in a new replacement task from the user.
 - 3.3.3. The Todo must be able to take in a new replacement datetime from the user.
 - 3.3.4. The Todo must be able to take in a new replacement duration from the user.
 - 3.3.5. The Todo must be able to take in a new replacement destination from the user.
 - 3.3.5.1. The website should update the route suggested for the old destination if the new destination differs from the old destination.
 - 3.3.6. The Todo must be able to take in a new replacement indoors/outdoors value from the user.
- 4. The website should have real-time access to weather data of the day.**
 - 4.1. If the forecasted weather of an uncompleted outdoor Todo is unideal (e.g. rainy, thunderstorms, hazy), then the website suggests the user to edit the Todo.
 - 4.1.1. The suggestion should show up as a notification on the main page.
 - 4.1.2. The website must provide the user an option to view alternative destinations by displaying an interactive location services map embed.
 - 4.1.2.1. The location services embed should display the map centered on the original Todo's destination.
 - 4.1.2.2. Users should be able to zoom in and out of the map.
 - 4.1.2.3. Users should be able to pan across the map to view different locations.
- 5. The website should have real-time access to the list of events happening near a location.**
 - 5.1. The user can choose an event from the list of events to view more information about the event.

- 5.1.1. The user must be able to add the event as a Todo.
 - 5.1.1.1. The title of the Todo must be the title of the event.
 - 5.1.1.2. The task of the Todo must be the details of the event.
 - 5.1.1.3. The datetime of the Todo must be the datetime of the event.
 - 5.1.1.4. The duration of the Todo must be the duration of the event.
 - 5.1.1.5. The destination of the Todo must be the destination of the event.
 - 5.1.1.6. If another Todo's datetime overlaps with the new Todo, the website displays a warning.

Non-functional Requirements

1. Performance

- 1.1. The website must load within 3 seconds.
- 1.2. The website must respond to user actions within 1 second.

2. Usability

- 2.1. More than 95% of users should be able to create a Todo within 10 minutes.
- 2.2. Users must be able to edit their Todo after creating one.
- 2.3. In-website FAQ guide must be available to address user questions related to the website's functionalities and location settings.

3. Compatibility

- 3.1. Map view must be clearly viewed, thus compatible with a variety of devices and screen sizes (e.g Mac, Windows, Linux etc).
- 3.2. Schedule view must be clearly viewed, thus compatible with a variety of devices and screen sizes (e.g Mac, Windows, Linux etc).

4. Privacy

- 4.1. The website must clearly inform users of the purpose of collecting location data.
- 4.2. The website must seek permission from users to collect location data.
- 4.3. Users must have the choice to disable location sharing at any time.
- 4.4. Users must have the choice to change their password.

5. Localization

- 5.1. Users must be able to choose their language of preference.
- 5.2. Map markers and distance units must be presented in the user's chosen language and regional settings.

6. Accessibility

- 6.1. The map interface must include appropriate labels and alternative text for screen readers.
- 6.2. Users with disabilities must be able to interact with the map effectively.

7. Accuracy

- 7.1. The information displayed should be accurate.
- 7.2. The events of Todo should be displayed in chronological order.
- 7.3. The events suggested by Todo should be displayed in close proximity to the user's intended location.

8. Support

- 8.1. Proper documentation must be provided for future developers.
- 8.2. The app must include help and feedback buttons for the users.

8.3. The app must provide FAQ for users that have enquiries.