API DOCUMENTATION TASK 2

**Personal Info**

*Name: Muneem Mushtaq*

*Ph: +917889910687*

[*Mail me here!*](mailto:emuneem06@gmail.com)

[*Link to my github*](http://github.com/Ehmadmuneem)

01.Creating Object Data Model For The Nudge

* The Nudge object should have the following properties:
* event\_id (string): Represents the ID of the tagged event.
* title (string): Specifies the title of the nudge.
* image (file): Allows the user to upload an image as the nudge cover.
* scheduled\_on ;(string): Represents the date of the nudge creationt.
* Timings:(object): “*from (string): Represents the starting time of the nudge. It is expected to be in a specific format, such as "HH:MM” to (string): Represents the ending time of the nudge. It is also expected to be in the format "HH:MM".*
* icon (string): Represents an icon associated with the nudge.”
* invitation (string): Specifies a one-line invitation for the nudge

Note: Data model file of nudges is present in the events project, root directory

02: Api Structure Documentation

Base URL: [Inserting base url]

API Endpoints:

**Create Nudge**:

* Endpoint: /nudges
* Request Type: POST
* Payload:
* event\_id (string): The uid of the tagged event.
* title (string): The title of the nudge.
* image (file): The uploaded image for the nudge cover.
* Scheduled\_on: (string): The scheduled date for sending the nudge.
* Timings(object): Timing of the event creation from-to.
* description (string): The description of the nudge.
* icon (string): The icon associated with the nudge.
* Workshop\_invitation (string): The one-line invitation for the nudge.
* Description: Allows users to create a new nudge for their event.

**Get Nudge:**

* Endpoint: /nudges/:nudgeId
* Request Type: GET
* Description: Retrieves information about a specific nudge identified by nudgeId.

**Update Nudge:**

* Endpoint: /nudges/:nudgeId
* Request Type: PATCH
* Payload: Any of the properties mentioned above (title, image, schedule\_time, description, icon, invitation) can be included in the request body to update the nudge.
* Description: Updates the details of a specific nudge identified by nudgeId.

**Delete Nudge:**

* Endpoint: /nudges/:nudgeId
* Request Type: DELETE
* Description: Deletes a specific nudge identified by nudgeId.

**Get All Nudges:**

* Endpoint: /nudges
* Request Type: GET
* Description: Retrieves a list of all nudges available.

02:Api CRuD Functionality Documentation

CRUD functionalities for the nudge feature using Express and MongoDB. Include details about each functionality, such as input parameters, expected responses, and any specific requirements.

Crud Functionality…

**Create Nudge:**

* Endpoint: POST /nudges
* Description: Allows users to create a new nudge for their event.
* Request Payload: Includes the required properties mentioned above in the API structure documentation.
* Response:
* Success: HTTP 201 (Created) with the newly created nudge ID.
* Error: HTTP 400 (Bad Request) if the request is invalid or incomplete.

**Get Nudge:**

* Endpoint: GET /nudges/:nudgeId
* Description: Retrieves information about a specific nudge identified by nudgeId.
* Response:
* Success: HTTP 200 (OK) with the nudge details.
* Error: HTTP 404 (Not Found) if the nudge is not found.

**Update Nudge:**

* Endpoint: PATCH /nudges/:nudgeId
* Description: Updates the details of a specific nudge identified by nudgeId.
* Request Payload: Includes any of the properties mentioned in the API structure documentation to update the nudge.
* Response:
* Success: HTTP 200 (OK) if the nudge is successfully updated.
* Error: HTTP 400 (Bad Request) if the request is invalid or incomplete.

**Delete Nudge:**

* Endpoint: DELETE /nudges/:nudgeId
* Description: Deletes a specific nudge identified by nudgeId.
* Response:
* Success: HTTP 200 (OK) if the nudge is successfully deleted.
* Error: HTTP 404 (Not Found) if the nudge is not found.

**Get All Nudges:**

* Endpoint: GET /nudges
* Description: Retrieves a list of all nudges available.
* Response:
* Success: HTTP 200 (OK) with an array of nudge details.
* Error: HTTP 404 (Not Found) if no nudges are found.