Time Wise

Features:

-Add Events with add edit delete functionality (1)(8)(13)

Task

Reminders

Notes (2)

Memories (2)

Meetings (3)

-Badges for all events (4)

-One Time Notification that summarises all schedule (5) which might have button open and see again.

-Login & Registration page

-User Profile Page (6)

//-Habit Journal View Page

-Right Panel including top 50 unassigned events (9)

Add Events from right pane to a date through drag and drop (7)

Add event button

-Left Panel shows items for the day and calendar for quick navigation (11)

-Middle Panel host calendar with events populated (12)

-Filtering options are based on month year type and tag (10)

-Different View Modes Month Week Day (12)

-Info area of the calendar being displayed (Newly Added) with a legend

Scope of each feature:

**Event**

Opens Up a pop up on click and displays respective details.

Allows Edit and deletion of events.

Deletion can only be done via pop up feature.

All events have priorities.

Each event has a type eg Task/,,,,

Each Event has a tag

Each Event has a title.

Sectioned by events and ordered by priority scrollable

On clicking this will open pop up.

Adding new event when date is specified goes to daysquare when null it goes to right pane.

**Task**

Task is an errand that you have to complete but before a time frame.

You can think of a task as an errand that you have to run whenever you are free.

In case if the task is not completed in due time it marks green or else red.

//or it automatically move to the next day.

Task appears on the left panel for the chosen day and checkable and has check box next to it.

Task cannot be deleted from check box.

Task Collection {

\_id: ObjectId,

title: String,

textBody: String,

dateCreated: DateTimeStamp,

dateAddedTo: DateTimeStamp

priority: Number,

tag: String,

expired: Boolean,

checked: Boolean

type: “task”

}

**Reminder**

Reminder throws alert when past a certain Datetime.

There can be recurring reminders - let's say schedule an appointment with a doctor. Using repeating counter Object below.

After DateTime crosses expired becomes true with UI change.

For reminders all date values are mandatory which means thats it does not got to right panel.

Reminder Collection {

\_id: ObjectId,

title: String,

priority: Number,

textBody: String,

tag: String,

dateCreated: DateTimeStamp,

dateAddedTo: DateTimeStamp

dateDueOn: DateTimeStamp

repeating: Boolean,

repeatingCounterIncrement: Number,

repeatingIncrementBy: String (Day| Week | Month | Year),,

expired:Boolean

repeatingGroup: ObjectId

type: “reminder”

}

**Notes**

Notes can allow us to store a piece of TXT/PDF/Images

Use Case- MOM for office/ CS 5?? Notes for that day etc

Instead of storing it in a folder which often is difficult to retrieve.

Note Collection {

\_id: ObjectId,

title: String,

dateCreated: DateTimeStamp,

dateAddedTo: DateTimeStamp

textBody: String,

tag: String,

documentLink: [String]

type: “notes”

}

**Memories**

This takes in a picture and text describing the picture and your feelings.

This can be thought of as a digitised personal diary.

This can be shareable but is out of scope for now

It allows only text entry as well of size 50 char or more need to decide upper limit

But it does not allow only pics to be uploaded to bring about habit of writing what you feel inside.

Tags to categorise memories. Happy Memory/SAD

Basically text editor inside notes.

**Meetings**

This event can help us focus our time on one particular activity. Brings about Focus

Use Case Meeting on zoom to discuss a project.

Preparation for exam

The event throws a alert when the meeting is about to start.

Meeting Collection {

\_id: ObjectId,

title: String,

dateCreated: DateTimeStamp,

dateAddedTo: DateTimeStamp,

dateDueOn: DateTimeStamp,

duration: String,

priority: Number,

textBody: String,

tag: String,

repeating: Boolean,

repeatingCounterIncrement: Number,

repeatingIncrementBy: String (Day| Week|Month),

expired:Boolean

repeatingGroup: ObjectId

type: “meeting”

}

**User Login & Registration**

**User Registration UI**

FirstName:

LastName:

Email:

Password:

Disability:

DOB: Should be greater than 13 |

Consent for data collection:

**Login UI**

Email

Password

DB Login & Registration

User Collection {

\_id: ObjectId,

first\_name: String,

last\_name: String,

email: String,

password: String,

Disability:boolean,

Dob:date,

Consent:boolean

taskIds: [ObjectId],

reminderIds: [ObjectId],

noteIds: [ObjectId],

meetingIds: [ObjectId]

}

List of activities:

**Backend**

**Jon: {**

**Login API**

**Registration API**

**Authorization API**

**Adding Authorization to routes**

**}**

Mahesh : {Task:

CUD operations

Read: GetALL(USER ID)

Get(TaskID)

}

Sunders :{Reminders:

CUD operations

Read: GetALL(USER ID)

Get(TaskID)

}

Sanjeet {Meetings:

CUD operations

Read: GetALL(USER ID)

Get(TaskID)

}

Sundera :{Notes:

CUD operations

Read: GetALL(USER ID)

Get(TaskID)

}

All CRUD will also update User collection by updating the respective array

Do we need an API to check time and update collection.

Display Attributes for each object type { color = badge = type/tag = } will be in constants.js file