

EVENT MANAGEMENT DASHBOARD

01. FRONTEND

EVENT MANAGEMENT

Upcoming Events

- localhost:3000: runing on local surver | Location: local | Date: 2024-12-22 Edit Delete
- second survor:4000: second survor | Location: local | Date: 2024-12-25 Edit Delete

Add New Event

Event Name:

Description:

Location:

Date: dd-mm-yyyy

Add Event

ATTENDEE MANAGEMENT

Attendee List

- 1 | Assigned Event: second survor :4000 Remove

Add Attendee

Enter attendee name:

Assign Event

Select Event:

Add Attendee

TASK TRACKER

Task Tracker

Tasks for localhost:3000

running on local surver | Due: 2024-12-22 | Assigned to: 1 Status: Completed

Task Description: second survor :4000

Task Deadline: 25-12-2024

Assign to Attendee:

Assign to Event:

Task Status: Pending

Add Task

COMPLETE FRONTEND

Event Management Dashboard

Upcoming Events

localhost:3000: runing on local surver | Location: local | Date: 2024-12-22

[Edit](#) [Delete](#)

Add New Event

Event Name

Description

Location

Date
 dd-mm-yyyy

[Add Event](#)

Attendee List

1 | Assigned Event: second survor :4000 [Remove](#)

Add Attendee
 Enter attendee name

Assign Event
 Select Event

[Add Attendee](#)

Task Tracker

Tasks for localhost:3000

running on local surver | Due: 2024-12-22 | Assigned to: 1 Status: Completed

Task Description
 second survor :4000

Task Deadline
 25 - 12 - 2024

Assign to Attendee
 Select Attendee

Assign to Event
 Select Event

Task Status
 Pending

[Add Task](#)

- **FRONTEND:** BUILT WITH HTML, CSS, AND JAVASCRIPT FOR DYNAMIC CONTENT AND STYLING.
- **BACKEND INTEGRATION:** USES THE FETCH API TO INTERACT WITH A SERVER (LIKELY NODE.JS/EXPRESS).
- **DATE HANDLING:** MANAGES EVENT AND TASK DEADLINES IN INDIAN STANDARD TIME (IST).

02. BACKEND

EVENT MANAGEMENT

CREATE AN EVENT ,
GET ALL EVENT

cluster0.s3xtw.mongodb.net > toDoP > events

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate](#) Explain Reset Find

[+](#) [Edit](#) [Delete](#)

25 1-2 of 2

```
_id: ObjectId('67681bbe1b3edf03caef5065')
title: "localhost 3000"
description: " runing on local surver"
location: "local"
date: 2024-12-22T00:00:00.000+00:00
__v: 0

_id: ObjectId('67681c461b3edf03caef5067')
title: "survor:4000"
description: " second survor"
location: "local"
date: 2024-12-25T00:00:00.000+00:00
__v: 0
```

now for the

UPDATE AN EVENT

ADD DATA EXPORT DATA UPDATE DELETE

```
_id: ObjectId('67681bbe1b3edf03caef5065')
title: "localhost 3000"
description: " runing on local surver"
location: "local"
date: 2024-12-22T00:00:00.000+00:00
__v: 0

_id: ObjectId('676828ee8d87e96fa3b40132')
title: "second survor:60000"
description: "second survorrr"
location: "not local"
date: 2024-12-25T00:00:00.000+00:00
__v: 0
```

DELETE AN EVENT

Documents 1 Aggregations Schema Indexes 1

Type a query: { field: 'value' } or [Generate](#) Explain Reset Find

[+](#) [Edit](#) [Delete](#)

```
_id: ObjectId('67681bbe1b3edf03caef5065')
title: "localhost 3000"
description: " runing on local surver"
location: "local"
date: 2024-12-22T00:00:00.000+00:00
__v: 0
```

ATTENDEE MANAGEMENT

The screenshot shows the MongoDB Compass interface for the attendees collection. The left sidebar lists collections: events, attendances, trackers, and a plus sign icon. The main area shows the 'attendances' collection with 1 document. The document details are:

```
_id: ObjectId('67684fe39273256c38ebd9be')
attendee : 1
assignEvent : "localhost:3000"
__v : 0
```

Below the document list are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE.

TASK MANAGEMENT

The screenshot shows the MongoDB Compass interface for the trackers collection. The left sidebar lists collections: events, attendances, trackers, and a plus sign icon. The main area shows the 'trackers' collection with 1 document. The document details are:

```
_id: ObjectId('6768225f8d87e96fa3b40130')
description : "localhost:3000"
date : 2024-12-22T00:00:00.000+00:00
attendance : 1
event : "localhost:3000"
status : "complete"
__v : 0
```

Below the document list are buttons for ADD, EDIT, and DELETE.

- 1. THE APPLICATION USES **EXPRESS.JS** TO MANAGE HTTP REQUESTS FOR EVENT, ATTENDANCE, AND TASK TRACKING APIs.
- 2. **MONGODB** STORES DATA WITH ***MONGOOSE* SCHEMAS** FOR EVENT(ADD,GET ALL,UPDATE,DELETE), ATTENDEE(ADD,GET ALL,DELETE), AND TRACKER(CREATE,GET ALL,UPDATE)MODELS.
- 3. **JWT AUTHENTICATION** ENSURES SECURE TOKEN-BASED USER ACCESS FOR EVENT CREATION AND EDITING OPERATIONS.

03. INTEGRATION

1. INTEGRATING THE FRONTEND WITH THE BACKEND APIs TO FETCH AND DISPLAY REAL-TIME DATA.

EVENT MANAGEMENT

The screenshot shows the Event Management application interface. On the left, there's a sidebar for "Upcoming Events" listing two events: "localhost:3000: running on local server | Location: local | Date: 2024-12-22" and "second survor :4000: second survor | Location: local | Date: 2024-12-25". Below this is a form to "Add New Event" with fields for Event Name (localhost:3000), Description (running on local server), Location (local), and Date (22 - 12 - 2024). A green "Add Event" button is at the bottom. On the right, there's an "Attendee List" section showing one assigned event (localhost:3000) and a form to "Add Attendee" with a placeholder "Enter attendee name". Below these is a "Connections" sidebar with a tree view of database connections, currently expanded to show "cluster0.s3xtw.mongodb.net" with various collections like E-commerce, admin, local, sample_mflix, test, and toDoP. The "events" collection is selected. A modal at the bottom left informs about Compass 1.45.0 availability.

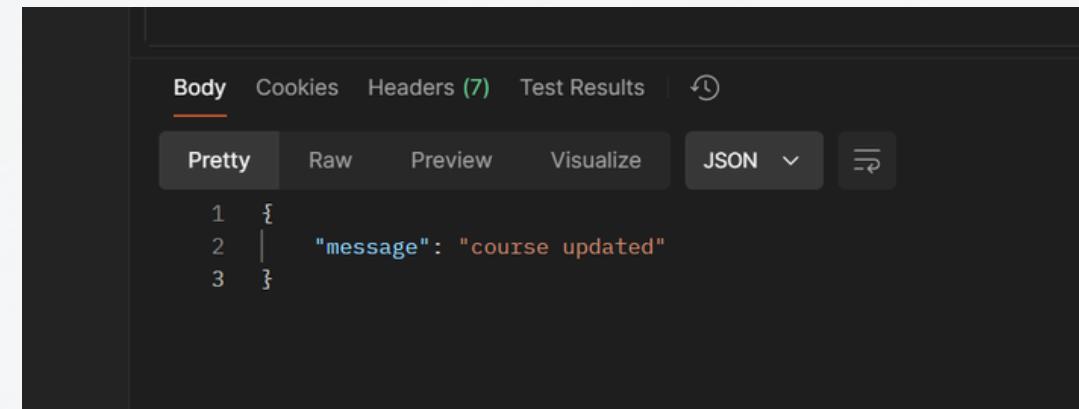
ATTENDEE MANAGEMENT

The screenshot shows the Attendee Management application interface. It features a "Event Management Dashboard" header. On the left, there's a "Upcoming Events" section listing the same two events as the first screenshot. To its right is an "Attendee List" section showing one assigned event (localhost:3000) and a form to "Add Attendee" with a placeholder "1". Below these are "Add New Event" and "Add Attendee" forms. On the right, there's a "Documents" list for the "attendances" collection, showing one document with fields: _id: ObjectId('67684fe39273256c38ebd9be'), attendee: 1, assignEvent: "localhost 3000", and --v: 0. There are buttons for "ADD DATA", "EXPORT DATA", and "UPDATE".

TASK TRACKER

The image shows two side-by-side screenshots. On the left is a user interface for a 'Task Tracker' application. It has fields for 'Task Description' (containing 'second survor :4000'), 'Task Deadline' (set to '25-12-2024'), 'Assign to Attendee' (set to '1'), 'Assign to Event' (set to 'second survor :4000'), and 'Task Status' (set to 'Pending'). A green 'Add Task' button is at the bottom. On the right is a MongoDB interface showing a single document in the 'Documents' tab. The document details are: _id: ObjectId('6768225f8d87e96fa3b40130'), date: "2024-12-22", event: "localhost:3000", taskdescription: "localhost:3000", status: "complete", attendee: "1". The interface includes tabs for Documents (1), Aggregations, Schema, Indexes (1), and Validation.

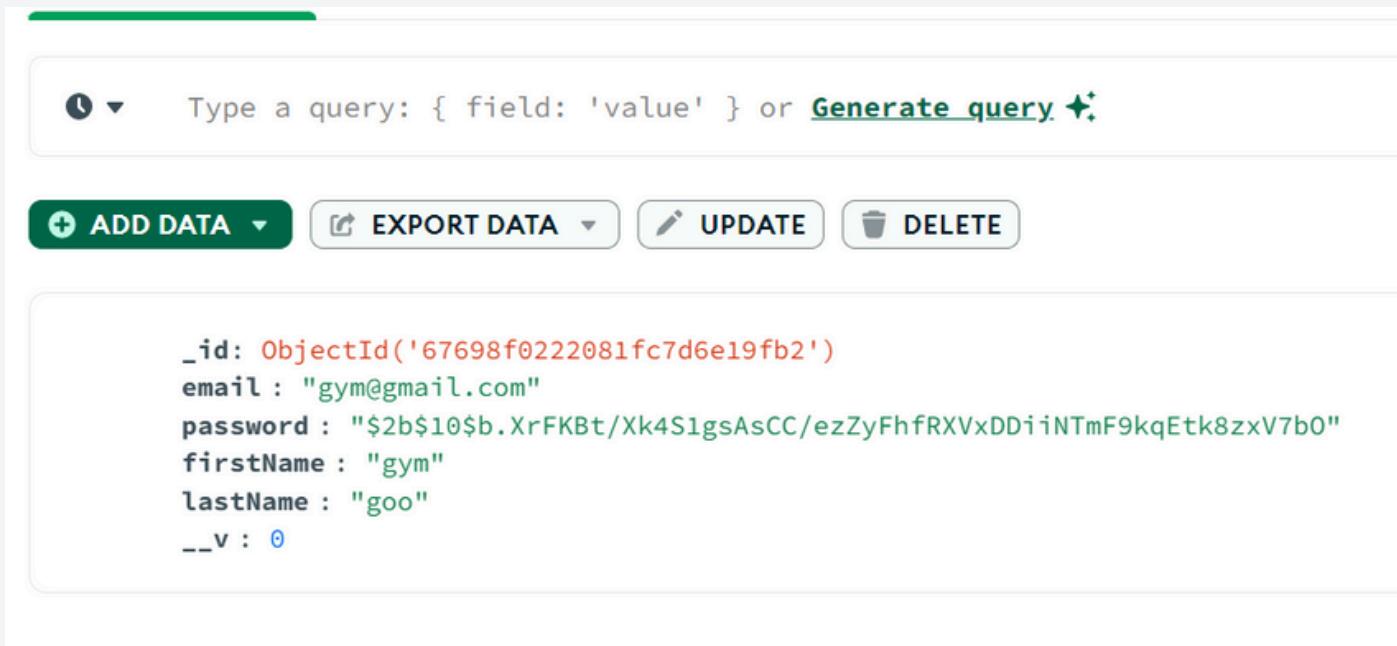
2. ENSURE SMOOTH USER INTERACTIONS



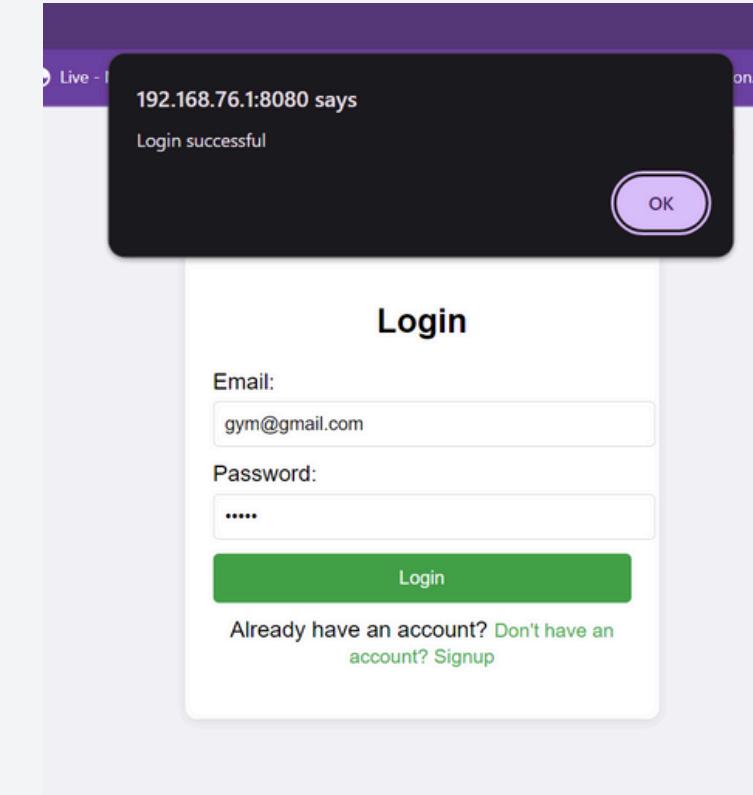
- **CORS CONFIGURATION:** CONFIGURES CORS TO ALLOW THE FRONTEND AT `HTTP://192.168.76.1:8080/HTMLV` TO ACCESS THE BACKEND API.
- **API ENDPOINTS:** PROVIDES `POST`, `PUT`, `DELETE` ENDPOINTS FOR CREATING AND MANAGING EVENTS, ATTENDANCE, AND TRACKERS.
- **JWT TOKEN USE:** SENDS A `JWT TOKEN` UPON EVENT CREATION TO AUTHORIZE SUBSEQUENT REQUESTS FROM THE FRONTEND.

BONUS POINTS

LOGIN/LOGOUT



```
_id: ObjectId('67698f0222081fc7d6e19fb2')
email : "gym@gmail.com"
password : "$2b$10$b.XrFKBt/Xk4S1gsAsCC/ezZyFhfRXVxDDiiNTmF9kqEtk8zxV7b0"
firstName : "gym"
lastName : "goo"
--v : 0
```



192.168.76.1:8080 says
Login successful

OK

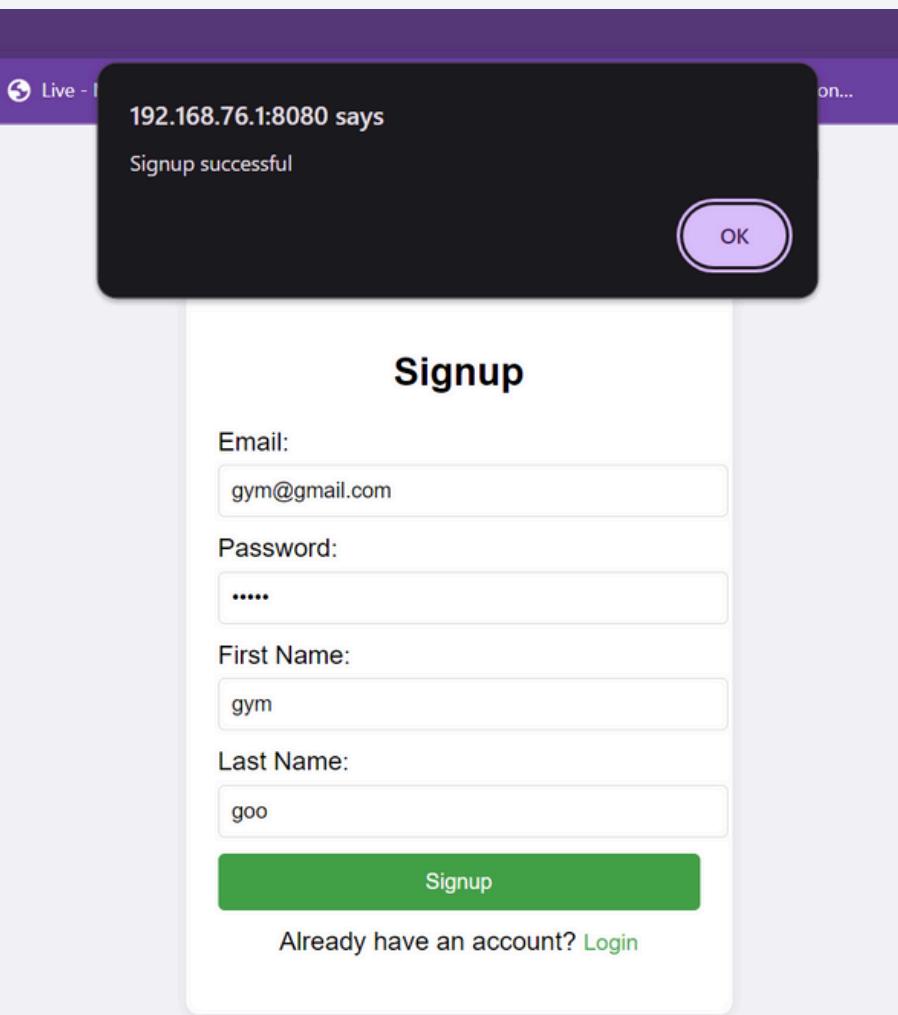
Login

Email:
gym@gmail.com

Password:
.....

Login

Already have an account? [Don't have an account?](#) [Signup](#)



192.168.76.1:8080 says
Signup successful

OK

Signup

Email:
gym@gmail.com

Password:
.....

First Name:
gym

Last Name:
goo

Signup

Already have an account? [Login](#)