Fitness Tracker

Database Management System Project

-by Gaurav Kumar

Presented by: Gaurav Kumar
Roll No : 2301331720023
Branch & Semester : Cyber Security (4th Sem)

Project Description

A web-based Fitness Tracker that allows users to input their Name and Day to retrieve fitness data, including steps taken, calories burned, workout details, and sleep patterns.

Technologies Used

- *- Frontend: HTML, CSS, JavaScript*
- *- Backend: Node.js, Express.js*
- *- Database: MongoDB*
- *- Other: Mongoose, Fetch API*

Features of the System

1. User Input & Data Retrieval

• Users can enter their Name and select a Day to fetch fitness data.

2. Fitness Data Management

• Stores and retrieves user fitness details, including steps, calories burned, heart rate, and active minutes.

3. Workout & Sleep Tracking

• Displays detailed workout information such as type, duration, distance, and calories burned.

• Tracks sleep data, including total sleep, deep sleep, REM sleep, and sleep score.

4. Dynamic Data Display

• Fetches and presents data in a well-structured modal popup for better readability.

5. Database Connectivity (MongoDB)

• Uses MongoDB to store fitness records and retrieve data efficiently.

System Architecture

1. Client-Side (Frontend) – User Interface

- *Built using **HTML, CSS, JavaScript**.*
- *Allows users to **input Name and Date** to fetch fitness data.*
- *Displays fitness data in a **modal popup**.*
- *Handles user input validation and error messages.*

2. Server-Side (Backend) – API & Business Logic

- *Developed using **Node.js and Express.js**.*
- *Handles **requests from the frontend** and processes them.*
- *Connects to the database using **Mongoose (MongoDB ODM)**.*
- *Provides a **REST API** to fetch user fitness data.*
- *Includes a **server health check endpoint**.*

3. Database Layer – MongoDB

- *Stores **fitness records for different users and days**.*
- *Uses a structured **User Schema** with Name, ID, and daily fitness logs.*
- *Retrieves and serves **fitness details based on user requests**.*

4. Data Flow in the System

1. ***User enters Name and Day** in the UI and clicks “Submit”.*
2. ***Frontend sends a request** to the Node.js backend via Fetch API.*
3. ***Backend processes the request**, queries MongoDB, and retrieves fitness data.*
4. ***Data is sent back to the frontend**, formatted, and displayed in a modal popup.*
5. ***If errors occur**, appropriate messages are shown to the user.*

Key Points in Database Design

1. **Users are uniquely identified by id**.

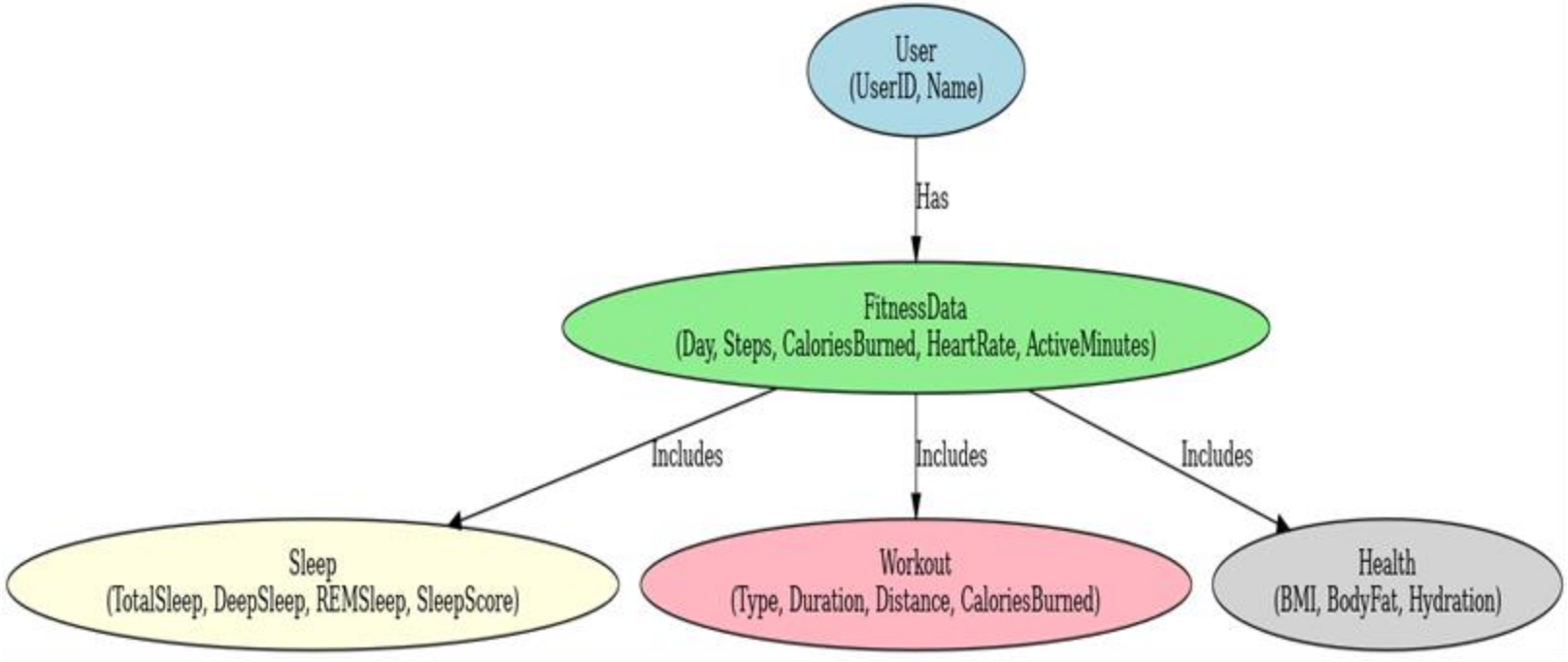
2. **Fitness data is stored in a days object**, where each day (Monday, Tuesday, etc.) has its own fitness stats.

3. **Nested structure** for sleep, workout, and health categories to keep the data structured.

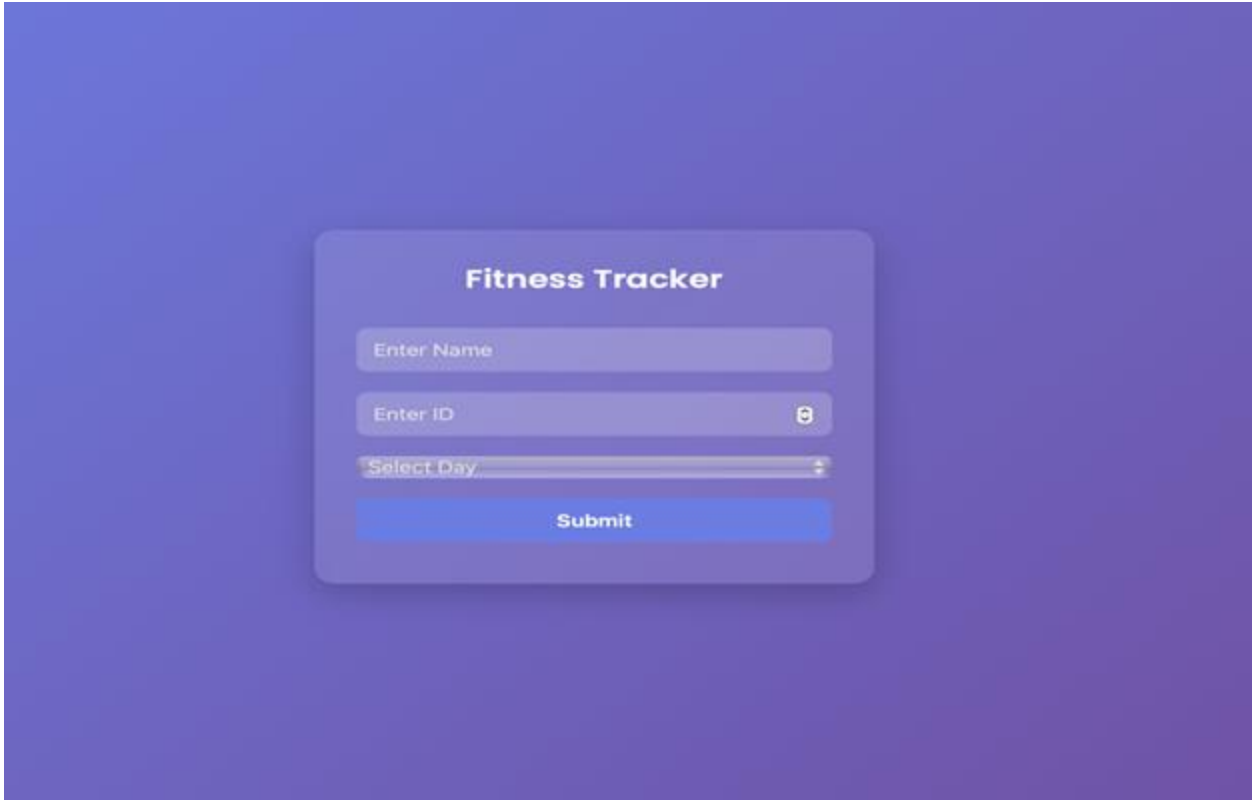
4. **Indexing id for faster lookup** in MongoDB.

5. **Scalable design**—new fields can be added easily without restructuring the database.

E-R Diagram


User
(UserID, Name)
Has
FitnessData
(Day, Steps, CaloriesBurned, HeartRate, ActiveMinutes)
Includes
Includes
Includes
Sleep
(TotalSleep, DeepSleep, REMSleep, SleepScore)
Workout
(Type, Duration, Distance, CaloriesBurned)
Health
(BMI, BodyFat, Hydration)

Screenshots of the Application



Test with given values

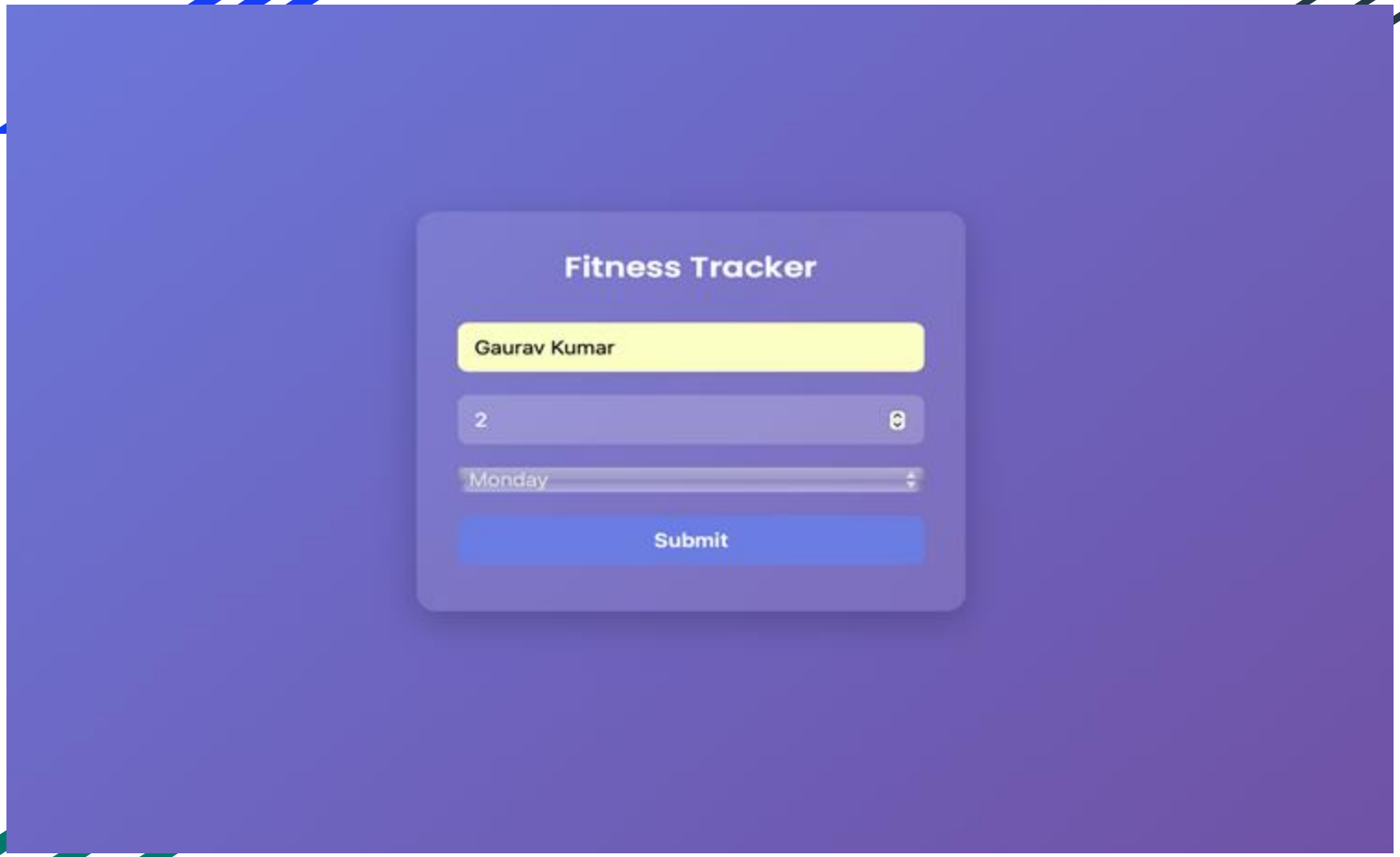
First make sure that server.js is running

```
_id: ObjectId('67c54aa5af72d2d69e77cf6a')
name : "Priyanshu Kumar"
id : 1
▸ days : Object

_id: ObjectId('67c54aa5af72d2d69e77cf6b')
name : "Gaurav Kumar"
id : 2
▸ days : Object

_id: ObjectId('67c54aa5af72d2d69e77cf6c')
name : "Lavesh Gaur"
id : 3
▸ days : Object

_id: ObjectId('67c54aecaf72d2d69e77cf6d')
name : "Aditya Palliwal"
id : 11
▸ days : Object
```


Fitness Tracker
Gaurav Kumar
2
8
Monday
Submit

Gaurav Kumar's Fitness Data (Monday)

×

Steps: 7800
Calories Burned: 1100
Heart Rate: 70 bpm
Active Minutes: 40

Sleep

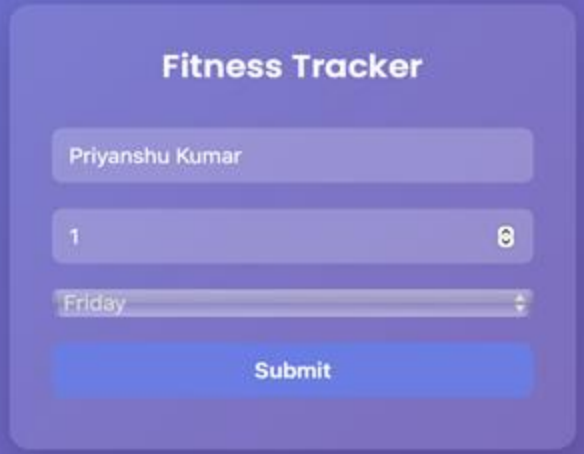
Total Sleep: 7h 10m
Deep Sleep: 2h 0m
REM Sleep: 1h 35m
Sleep Score: 82

Workout

Type: Walking
Duration: 30 mins
Distance: 3 km
Calories Burned: 200

Health

BMI: 23
Body Fat: 19%
Hydration: 55%


Fitness Tracker
Priyanshu Kumar
1
8
Friday
Submit

Priyanshu Kumar's Fitness Data (Friday)

×

Steps: 8700
Calories Burned: 1250
Heart Rate: 74 bpm
Active Minutes: 48

Sleep

Total Sleep: 7h 35m
Deep Sleep: 2h 12m
REM Sleep: 1h 47m
Sleep Score: 86

Workout

Type: Yoga
Duration: 40 mins
Distance: 0 km
Calories Burned: 150

Health

BMI: 22.5
Body Fat: 18%
Hydration: 62%

Source code and Dependencies

For source code you can visit my GitHub Repo just following by given link:

https://github.com/orewagaurav/mongodb-nodejs

Challenges Faced !

- *- Ensuring real-time database connectivity*
- *- Handling user input validation effectively*
- *- Managing asynchronous API requests*
- *- Implementing a responsive design for different screen sizes*

Conclusion

*The **Fitness Tracker Project** is a well-structured **web-based application** designed to efficiently store, manage, and retrieve user fitness data. By leveraging **MongoDB as the database, Node.js with Express.js for backend operations, and a modern frontend with HTML, CSS, and JavaScript**, the system ensures a **seamless and responsive user experience**.*

*Key features such as **real-time data fetching, workout and sleep tracking, user input validation, and a dynamic UI** make it a **robust and scalable solution**. The integration of **server health checks and error handling** further enhances its reliability.*

*This project can be **expanded in the future** by adding **user authentication, interactive analytics dashboards, smartwatch API integration, and real-time health monitoring**. Overall, the **Fitness Tracker Project** successfully demonstrates the power of **full-stack development** in managing fitness data effectively.*

Thank You !