HABIT TRACKER

A PROJECT REPORT

submitted by

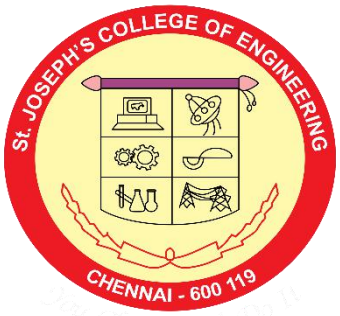
Joshitha A N 312323205102

Of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF INFORMATION TECHNOLOGY



ST. JOSEPH'S COLLEGE OF ENGINEERING

(An Autonomous Institution)

St. Joseph's Group of Institutions

OMR, Chennai- 600119

ANNA UNIVERSITY: CHENNAI

NOVEMBER-2024

ANNA UNIVERSITY: CHENNAI- 600025

GITHUB

Abstract

The Habit Tracker is a personal productivity tool designed to help users monitor their habits, stay motivated, and build positive routines. The system allows users to create, track, and analyze various habits over time. By consistently logging daily activities, users gain insights into their progress and improve their habits, aiming for long-term lifestyle changes.

Introduction

In today's busy world, forming and maintaining positive habits can be challenging. Many people struggle to keep track of their routines, which is where a habit tracker comes in. A habit tracker enables users to monitor their daily habits, track streaks, and visualize their progress, fostering self-discipline and goal-setting.

Objective

The primary objective of this project is to develop a habit tracking application that:

- Allows users to create and manage habits.

- Provides visual analytics to monitor progress.

- Encourages users through streaks and reminders to build consistent routines.

Scope of the Project

The scope of this project includes:

- Development of a web or mobile-based habit tracking system.

- Support for multiple habits and customizable tracking intervals.

- A user-friendly interface to visualize habit progress.

- Features like reminders, habit streaks, and completion notifications.

Technologies Used

*1. **Frontend**: HTML, CSS, JavaScript, React or Vue (for a web-based application)*

*2. **Backend**: Node.js with Express or Django (for handling data and API)*

*3. **Database**: MongoDB, Firebase, or SQLite for habit and user data storage*

*4. **Authentication**: Firebase Auth or Auth0 (for user login and registration)*

*5. **Visualization**: Chart.js or D3.js for habit tracking charts and graphs*

Project Description

The Habit Tracker allows users to log in, create new habits, and set goals for each habit. Each habit can be customized by the user to track specific

frequencies, such as daily, weekly, or monthly. As users log their daily progress, they can view streaks, receive reminders, and check analytics on their performance. The system aims to motivate users by showing their habit-building patterns over time, ultimately supporting lifestyle improvements.

Source Code

Here's a basic structure for setting up a habit tracker project.

*1. **Frontend**:*

- User Interface (React or Vue) for habit logging and progress display.

- CSS or Bootstrap for styling.

*2. **Backend**:*

- RESTful APIs with Express (Node.js) to handle habit creation, updating, and tracking.

- Database schema for habit and user data.

*3. **Database**:*

- Structure for users, habits, and progress logs.

Example Code Snippets

Backend - Node.js with Express (API for Creating Habits)

```javascript
const express = require('express');
const mongoose = require('mongoose');
```

```
const Habit = require('./models/Habit'); // Habit model schema

const app = express();
app.use(express.json());

app.post('/api/habits', async (req, res) => {
  try {
    const habit = new Habit({
      name: req.body.name,
      frequency: req.body.frequency,
      userId: req.body.userId,
    });
    await habit.save();
    res.status(201).json(habit);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
});

mongoose.connect('mongodb://localhost:27017/habitTracker', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
app.listen(3000, () => console.log('Server running on port 3000'));
```

Frontend - React Component for Habit Tracking Form

```javascript
import React, { useState } from 'react';

const HabitForm = ({ addHabit }) => {
  const [habit, setHabit] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    addHabit(habit);
    setHabit('');
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        value={habit}
        onChange={(e) => setHabit(e.target.value)}
        placeholder="Enter a new habit"
      />
      <button type="submit">Add Habit</button>
    </form>
  );
```

};

export default HabitForm;

```

## *System Design*

*1. **Database Design**: User table, Habit table (habit name, frequency, target, streaks).*

*2. **Frontend Design**: Interactive dashboard with a form to log habits, view streaks, and set reminders.*

*3. **Backend**: API endpoints for CRUD operations (Create, Read, Update, Delete) for habits.*

## *Key Features*

*1. **Habit Creation**: Users can define new habits with desired goals and frequencies.*

*2. **Progress Tracking**: Users log daily entries, with options to mark as complete or incomplete.*

*3. **Visual Analytics**: Charts to visualize progress over time.*

*4. **Reminders and Notifications**: Users receive reminders to help maintain habit consistency.*

*5. **Streak Tracking**: Users see streaks to stay motivated.*
```

Code Implementation

You can implement the above code in steps:

*1. **Database Setup**: Define models for users and habits.*

*2. **API Development**: Build RESTful endpoints.*

*3. **Frontend Components**: Set up forms and UI elements to capture user data.*

*4. **Integrate Charts**: Use libraries like Chart.js to display analytics.*

Future Enhancements

*1. **Gamification**: Add rewards and levels to increase user engagement.*

*2. **Social Sharing**: Allow users to share their progress with friends.*

*3. **Advanced Analytics**: More in-depth insights, habit correlation, and breakdowns.*

*4. **AI-based Suggestions**: Provide personalized habit recommendations based on user data.*

Conclusion

The Habit Tracker project is an essential tool for individuals looking to build and maintain positive habits. By tracking progress visually and through reminders, the application promotes a disciplined lifestyle, helping users achieve their personal development goals.

References

*1. **Chart.js Documentation**:
https://www.chartjs.org/docs/latest/*

*2. **Express.js Documentation**:
https://expressjs.com/*

*3. **React Documentation**: https://reactjs.org/docs/getting-started.html*

*4. **MongoDB Documentation**:
https://docs.mongodb.com/*