MERN CIPHER-SCHOOL

Smart Study Planner

PROJECT DOCUMENTATION

Submitted By
Udit

Registration No. 12307116

Programme : B.Tech CSE

Under the Guidance of
Mr. Pratik Raj

Introduction :

The **Smart Study Planner** is a simple web application developed to help students manage their daily study tasks, subjects, and schedules efficiently.
The main goal of this project was to understand the basics of **HTML, CSS, and JavaScript** while creating a functional and user-friendly website without using any frameworks or external libraries.

Objectives :

The objectives of building this project were:

- To practice frontend web development fundamentals.
- To understand **DOM manipulation** using JavaScript.
- To learn how **LocalStorage** works for saving data in the browser.
- To design a simple and clean **User Interface (UI)**.
- To build a project that is easy to understand and maintain.

Design :

I used Figma to design it and took the dimensions and other ideas from there.

The color code that I used was from online references.

COLOR CODE:

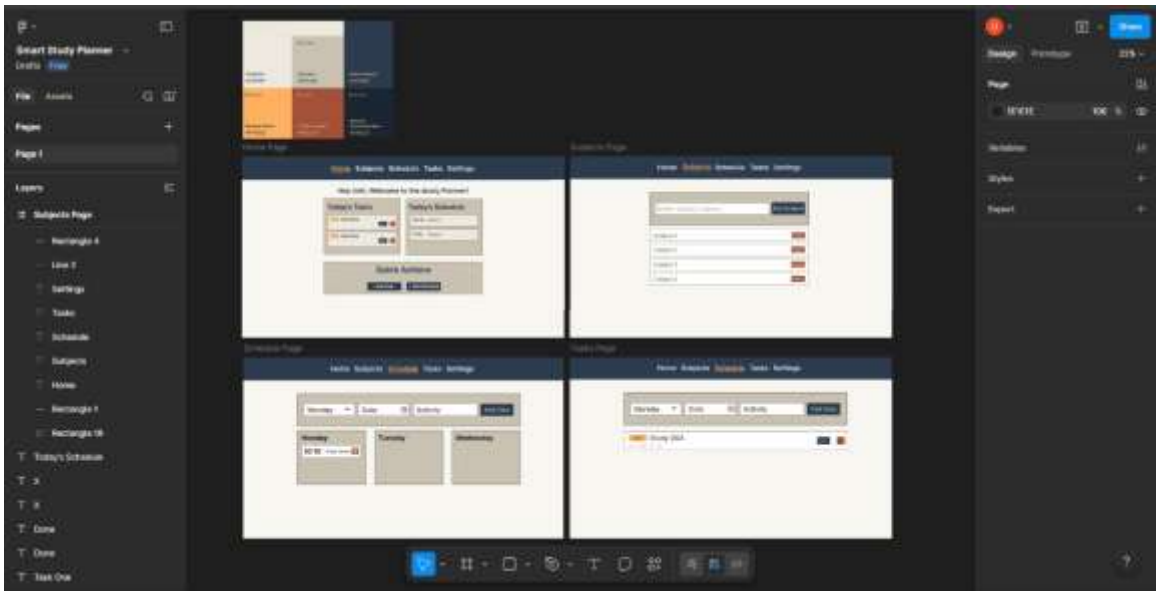
Using these colors and very simple and basic designs on Figma I was able to understand my needs in the style.css file.

I made 4 different pages in Figma :

1) Home Page :
   In this the main requirement was of 3 Divs where I can see the tasks, schedules and the quick action tab to quickly add tasks.
2) Subject Page:
   As the name suggests, this was the page used to add the subjects simply.
3) Schedule Page:

Here I can properly add my schedules and even check the different schedules already present

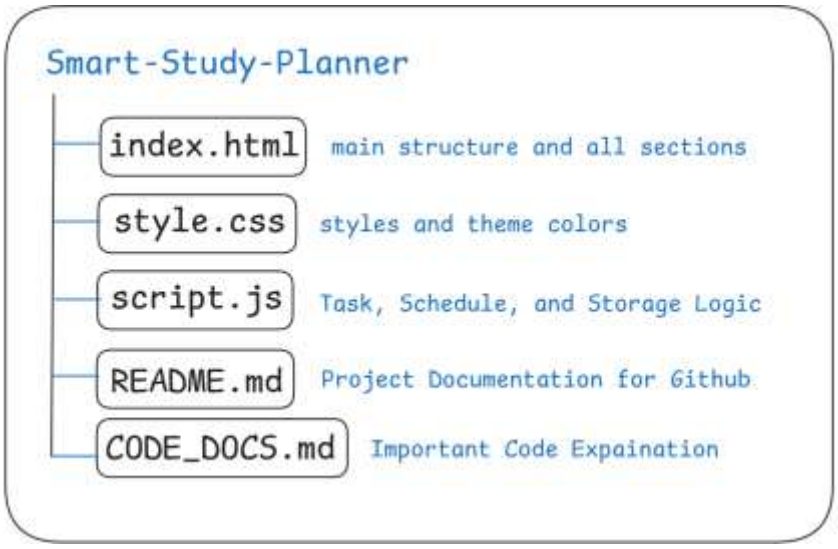
4) Tasks Page
5) Settings



Figma Design Board Image

File Structure:

I wanted to keep the file structure very simple, although it ended up creating problems for me during changing and finding the specific code parts. So, I added comments in order to find chunks and pieces of my code easily.
The main file structure is simple.



Script (JavaScript Implementation) :

The JavaScript file (script.js) is responsible for handling all the logic, interactivity, and data storage of the Smart Study Planner.
The entire application works dynamically using JavaScript without any backend or external libraries.

Data Storage using LocalStorage

To make sure that user data does not get lost after refreshing or closing the browser, I implemented **LocalStorage**.

LocalStorage works like a small internal database inside the browser where data is stored in **Key–Value pairs**.

The following keys were used:

- studyPlannerNames – Stores the list of subjects.
- studyPlannerTasks – Stores all the tasks added by the user.
- studyPlannerSchedule – Stores weekly schedule data.
- studyPlannerTheme – Stores the selected theme (Light/Dark).

Whenever the user adds or deletes information, the updated data is automatically saved. This ensures **data persistence** without requiring any server or login system.

Main JavaScript Functions

The JavaScript code is divided into small reusable functions so that the logic remains simple and easy to understand.

1. Data Handling Functions

- **loadData()**
  This function runs when the website opens. It checks if any saved data exists in LocalStorage and loads it into arrays so it can be displayed on the screen.
- **saveData()**
  This function runs whenever a change occurs (adding or deleting tasks, subjects, or schedules).
  It converts the data into a string format using JSON.stringify() and stores it in LocalStorage.
- **resetData()**
  Clears all saved information from LocalStorage and refreshes the page to start from a clean state.

2. Navigation Functions

- **showSection(sectionId)**
  Controls page navigation. It hides all sections and only shows the selected section when a navigation tab is clicked.
  It also highlights the active tab in the navbar.

- **goToAddTask() / goToAddSchedule()**
  These are helper functions used in the Home dashboard's quick actions. They automatically open the required page and pre-fill the current date or day for convenience.

3. Subject Management

- **addSubject()**
  Takes input from the user and adds it to the subject list. After adding, the data is saved and displayed instantly.

- **renderSubjects()**
  Loops through the subject array and dynamically creates list elements (<li>) in HTML.
  It also updates the subject dropdown menu in the Task section.

4. Task Management

- **addTask()**
  Creates a new task object containing:
    - Unique ID
    - Task Title
    - Subject
    - Date
    - Completion Status

- **renderTasks()**
    - Sorts tasks according to date.
    - Displays all tasks in the Tasks tab.
    - Filters and shows only **today's tasks** on the Home Dashboard.

5. Schedule Management

- **addSchedule()**
  Allows users to add classes or activities for specific days.

- **renderSchedule()**
  Groups schedule entries by weekdays and displays them properly.
  It also shows the current day's schedule on the dashboard.

Styling Interaction with JavaScript

JavaScript also interacts with CSS for theme switching:

- CSS Variables are defined in style.css.
- When the user switches theme, a class called .dark-theme is added or removed from the <body> element.
- This changes color variables instantly without reloading the page.

Hyper Links :

Github Repository Link : Smart Study Planner Github

Hosted Website Link : Smart Study Planner Website

Figma Design Link : Smart Study Planner Figma