

UpNext

**- A Smart Productivity and Study Companion for
Students Featuring AI-Driven Task Management and
Notes Organization**

Team Members & Team Guide

LITHIKA S
RA2231005010011

HARINI S
RA2231005010015

NISHA SHARMA R
RA2231005010052

Guide Name
Dr.Meenakshi .S

Introduction

The idea for this Productivity Web App came from the need to help students organize their tasks, improve productivity, and manage their time effectively. As a student, keeping track of multiple assignments, deadlines, and personal tasks can be overwhelming. This app is designed to simplify task management by allowing students to create tasks, set priorities, and allocate time efficiently.

One of its key features is a centralized notes section, where students can store all their important information in one place for easy access. By integrating task management, prioritization, and note-keeping in a single platform, the app aims to enhance productivity and ensure students stay on top of their responsibilities without feeling overloaded.



Problem Statement

01.

Students often struggle with managing tasks, deadlines, and study materials efficiently, leading to disorganization, missed deadlines, and reduced productivity.

02.

One of the major issues is the lack of a centralized notes management. Many students rely on multiple tools and applications to manage tasks and notes, which leads to fragmented information spread across different platforms.

03.

Another key challenge is inefficient time management. Without smart scheduling and prioritization, students often struggle to balance multiple assignments, projects, and personal tasks. This results in procrastination, last-minute stress.

Existing System



01

In today's digital landscape, several applications like Todoist, Google Tasks, Notion AI, and ChatGPT cater to productivity and task management. While these tools offer features such as task tracking, note-taking, and AI assistance, they often serve a general audience and lack personalization for students.



02

Notion AI offers flexibility and AI-powered writing assistance, yet it comes with a steep learning curve and is not tailored to student workflows. ChatGPT, though powerful, exists separately from productivity tools and lacks integration with real-time user data.

Proposed System



01

Our proposed MERN stack-based productivity web application bridges this gap by combining essential features like task management, deadline tracking, file uploads (via Cloudinary), and an integrated AI chat assistant—all within a clean, student-centric interface.



02

By combining intelligent, context-aware support with features like deadline tracking, file uploads, and a user-friendly interface, the application goes beyond basic task management to foster better time management, goal setting, and overall productivity.

Specifications

Hardware Specifications

Processor : Intel Core i5 / AMD Ryzen 5.

Memory : 8 GB RAM

Storage : 50GB free space

System Type : 64-bit operating system

Display : A standard monitor without pen or touch input

Graphics: Integrated graphics sufficient (No dedicated GPU required unless using heavy AI models locally)

Software Specification

Operating Systems

Windows 10/11
macOS (Monterey or later)
Linux (Ubuntu 20.04 or later)

Backend

Runtime Environment: Node.js
Framework: Express.js
Language: JavaScript

Frontend

Language: JavaScript, HTML5, CSS3
Library: React.js
Styling Framework: Bootstrap
Icons/Assets: Font Awesome,
Custom SVGs

Database:

MongoDB
(Cloud-hosted via MongoDB Atlas)

System Architecture

Frontend (User Interface & Client)

- **React.js** – For building a dynamic and interactive UI
- **Bootstrap** – For styling and responsive design

Backend (Server & API)

- **Node.js with Express.js** – For handling API requests and business logic
- **Multer** – For handling file uploads (e.g., storing attachments or notes)

Database (Data Storage)

- **MongoDB (NoSQL Database)** – For storing tasks, notes, and user data

External Services

- **Cloudinary**: For image and file uploads, storing media assets securely in the cloud
- **AI API** (e.g., OpenAI): For AI-based assistant functionality

Database Description

The Database Layer is responsible for securely storing and managing all essential application data, ensuring efficient retrieval and integrity. **MongoDB** is used as the primary database

User Schema

Stores user information, including usernames, hashed passwords, roles, and authentication details (Includes user id, created time, email, password and user name)

Task Schema

Maintains task-related information, including task descriptions, deadlines, priorities, and user assignments (contains title, description, priority, tags, status)

Reminder Schema

Tracks reminders linked to tasks, with timestamps and notification settings. (contains title, timestamp for each)

Files Management Schema

Manages file uploads, storing metadata such as file paths, types, and associated user references (folder and file names, paths, parent folder, created on etc.)

Chat Schema

Manages messages, user information, timestamps of messages and everything. (contains messages, time stamps, response from API)

Modules Description

O1. User Authentication Module

- Handles user registration, login, and logout
- Implements secure authentication using JWT
- Validates user credentials and manages session flow

O2. Task Management Module

- Allows users to create, update, delete, and view tasks
- Supports task details like title, deadline, status, and priority

O4. Dashboard Module

- Displays a user-friendly overview of tasks and upcoming deadlines
- Shows task summaries, reminders, and progress

O3. AI Chat Assistant Module

- Integrated using OpenAI API (or similar)
- Provides personalized suggestions, productivity tips, and answers
- Acts as a virtual study companion to boost motivation

Modules Description

05. File Upload Module

- Handles uploading of documents, images, or notes
- Uses Cloudinary to store and retrieve file URLs
- Allows students to attach resources to note

06. Database Module

- Manages interaction with MongoDB Atlas
- Uses Mongoose schemas for collections like Users, Tasks, Notes
- Handles CRUD operations securely and efficiently

Task Management

- Create Tasks: Users can add tasks with a title, description, and due date.
- Edit & Delete Tasks: Modify or remove tasks as needed.
- Task Categories: Organize tasks into different categories (e.g., Assignments, Exams, Personal Tasks).

Task Prioritization & Time Management

- Set Priorities: Tasks can be labeled as High, Medium, or Low priority.
- Deadline Reminders: Notifications or visual indicators for upcoming deadlines.

Notes Management

- Create & Store Notes: Users can write and save important notes.
- Create folders and sub folders for your notes.
- File Upload (Multer Integration): Attach PDFs, images, or other files for reference.

Conclusions & Future Enhancement

User Authentication & Security

- Sign Up & Login: Secure user registration with JWT authentication.
- Password Encryption: Ensuring data security using bcrypt.js.
- Register and login with email and password

Responsive & User-Friendly Design

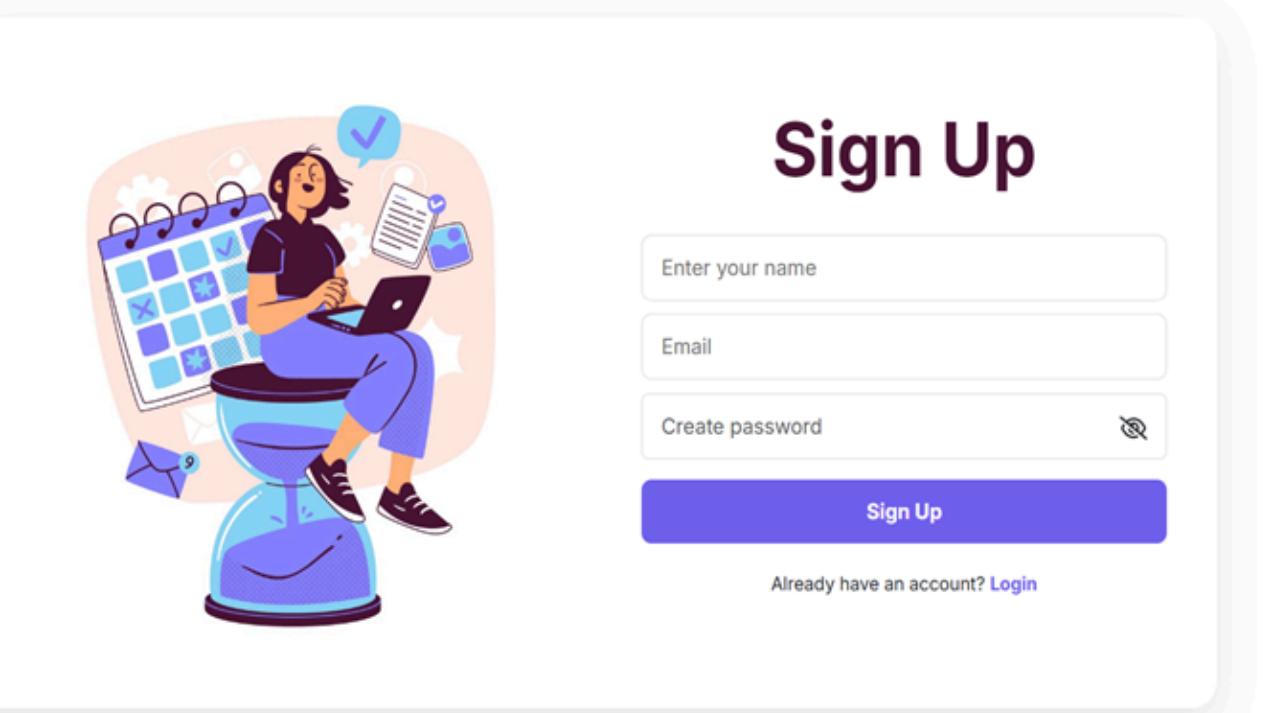
- Mobile & Desktop Compatibility: Works seamlessly on different screen sizes.
- Minimalist UI: Designed for efficiency without unnecessary distractions.

AI-Powered Task Guidance for Students

- When a student enters a task title, AI analyzes it and suggests step-by-step actions to complete it efficiently.
- AI suggests optimal time allocation based on task complexity and urgency.

OutPut Screen

Sign Up Page



Sign Up

Enter your name

Email

Create password
 

Sign Up

Already have an account? [Login](#)

Login page



Login

Email

Password
 

Login

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

OutPut Screen

Dashboard

UpNext

Menu

- [Home](#)
- [Tasks](#)
- [Notes](#)
- [Ask AI](#)
- [Calendar](#)

[Logout](#)

Home

Search task

HariniS H

Good Afternoon, HariniS

Sunday, 6th April 2025

Total tasks 3

To-do tasks 1

In-progress tasks 1

Completed tasks 1

My Tasks

- Compare AWS vs. Azure
- Solve Python Problems on LeetCode
- Create Personal Portfolio Website

My Reminders

- submit python assignment 5:00 PM edit trash
- register company 7:00 PM edit trash
- Submit assignment 6:00 PM edit trash

AI

OutPut Screen

Task page

UpNext

Menu

- Home
- Tasks**
- Notes
- Ask AI
- Calendar

Logout

Tasks

Search task

HariniS H

All Tasks

Add Task

TO DO

High

Compare AWS vs. Azure

Create a comparison chart of features, pricing, and use cases for cloud platforms....

23rd Jan

IN PROGRESS

Low

Solve Python Problems on LeetCode

Complete 5 beginner-level problems focusing on strings and arrays....

23rd Jan

COMPLETED

Low

Create Personal Portfolio Website

Use HTML, CSS, and JavaScript to build a basic portfolio site with a responsive design....

23rd Jan

OutPut Screen

Notes page

The screenshot shows a notes application interface. On the left, a sidebar titled "UpNext" contains links: "Menu", "Home", "Tasks", "Notes" (which is highlighted in blue), "Ask AI", and "Calendar". At the bottom of the sidebar is a "Logout" button. The main area is titled "Notes" and has a search bar labeled "Search task" with a magnifying glass icon. In the top right corner, there is a user profile for "HariniS" with a blue circular icon containing a white letter "H". Below the title, the text "My Notes" is displayed in bold. Underneath it, the path "Root >" is shown. A list of items is displayed in a grid: "newww" (a folder icon), "New folder" (a folder icon), "Questions.pdf" (a PDF icon), and another "Questions.pdf" (a PDF icon). Each item has a small red trash can icon to its right. In the top right of the main area, there are two buttons: "+ New Folder" and "Upload File".

OutPut Screen

Ai

The screenshot shows the UpNext application interface. On the left is a sidebar with a purple header "UpNext" and a "Menu" section containing "Home", "Tasks" (which is highlighted in purple), "Notes", "Ask AI", and "Calendar". At the bottom of the sidebar is a "Logout" button. The main area has a header "Tasks" with a search bar and a user name "HariniS". Below the header is a sub-header "All Tasks" and a "TO DO" section. In the "TO DO" section, there is a task titled "Compare AWS vs. Azure" with a priority of "High", a due date of "23rd Jan", and a description: "Create a comparison chart of features, pricing, and use cases for cloud platforms....". To the right of the tasks is a floating chat window titled "UpNext Assistant" with the identifier "A". The chat window shows a conversation where the user says "hi" and the AI assistant responds with: "Hello! How can I help you today? If you have any questions or need assistance with something, feel free to ask. I'm here to help. If you just want to chat, we can do that too. What's on your mind?". The user then asks "Can you explain what react does" and the AI assistant responds with: "React is a popular JavaScript library for building user interfaces, particularly for single-page applications. It's used for building". At the bottom of the chat window is a text input field "Type your message here" and a "Send" button.

References

React.js Documentation: <https://react.dev/>

Bootstrap- [Bootstrap Documentation](#)

Node.js Documentation: <https://nodejs.org/en/docs>

Express.js Documentation: <https://expressjs.com/>

MongoDB Documentation: <https://www.mongodb.com/docs/manual/>

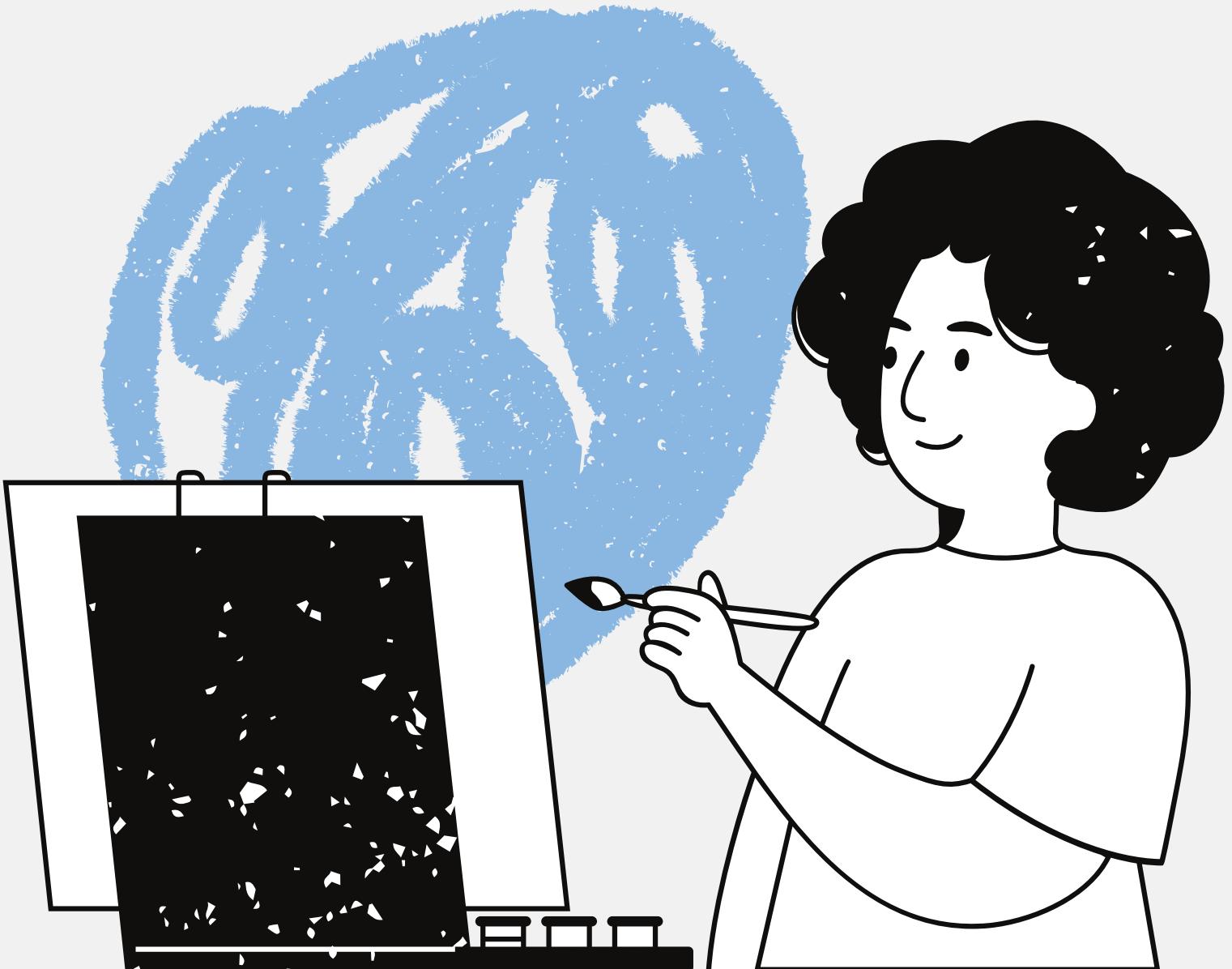
Stack Overflow: <https://stackoverflow.com/>

MDN Web Docs: <https://developer.mozilla.org/en-US/>

GeeksforGeeks: <https://www.geeksforgeeks.org/>

Final reflections and future steps

Building this Productivity Web App has been a rewarding experience, as it addresses a real problem students face – managing tasks, prioritizing work, and improving productivity. Through the MERN stack, AI integration, and intuitive UI, we have created a tool that simplifies student life by keeping everything in one place. The project not only strengthened our technical skills but also taught us valuable lessons in team collaboration, problem-solving, and user-centric design.



**Thank you
very much!**