



Vivekanand Education Society's Institute Of Technology

Department Of Information Technology

DSA mini Project

A.Y. 2025-26

Title: AI-Based Priority Todo list

Sustainability Goal :Promotes productivity and time management through AI-based prioritization.

Domain:Artificial Intelligence (AI) and
Productivity Enhancement

Member:Tanay Patil

Mentor Name: Kajal Joseph

1 NO
POVERTY



2 ZERO
HUNGER



3 GOOD HEALTH
AND WELL-BEING



4 QUALITY
EDUCATION



5 GENDER
EQUALITY



6 CLEAN WATER
AND SANITATION



7 AFFORDABLE AND
CLEAN ENERGY



8 DECENT WORK AND
ECONOMIC GROWTH



9 INDUSTRY, INNOVATION
AND INFRASTRUCTURE



10 REDUCED
INEQUALITIES



11 SUSTAINABLE CITIES
AND COMMUNITIES



THE GLOBAL GOALS
For Sustainable Development

12 RESPONSIBLE
CONSUMPTION
AND PRODUCTION



13 CLIMATE
ACTION



14 LIFE BELOW
WATER



15 LIFE
ON LAND



16 PEACE AND JUSTICE
STRONG INSTITUTIONS



17 PARTNERSHIPS
FOR THE GOALS





Content

1. Introduction to the Project
2. Problem Statement
3. Objectives of the Project
4. Scope of the Project
5. Requirements of the System (Hardware, Software)
6. ER Diagram of the Proposed System
7. Data Structure & Concepts Used
8. Algorithm Explanation
9. Time and Space Complexity
10. Front End
11. Implementation
12. Gantt Chart
13. Test Cases
14. Challenges and Solutions
15. Future Scope
16. Code
17. Output Screenshots
18. Conclusion
19. References (in IEEE Format)



Introduction to Project

The AI-Based Priority To-Do List is an intelligent task-management web application that combines Data Structures and Algorithms (DSA) with Artificial Intelligence (AI) to help users organize and prioritize their daily tasks efficiently.

It automatically analyzes each task using AI text understanding and assigns:

- Priority levels (High, Medium, Low)
- Categories (Work, Personal, Health, etc.)
- Deadlines



Problem Statement

Managing daily tasks manually often leads to confusion and missed priorities. Users need a smart system that can automatically organize tasks based on importance and deadlines — reducing stress and improving productivity.



Objectives of the project

- To develop an AI-powered To-Do List that automatically prioritizes and categorizes user tasks.
- To apply Data Structure concepts like Priority Queue, HashMap, and Stack for efficient task management.
- To integrate NLP using Hugging Face API for smart understanding of task descriptions.
- To provide real-time summaries and insights on task completion and daily productivity.
- To create a user-friendly and intelligent system that combines AI automation with DSA efficiency.



Requirements of the system (Hardware, software)

Software Requirements:

Operating System: Windows 10 / 11, macOS, or Linux
Frontend:

- React.js, CSS3
- Node.js (for npm)

Backend:

- Node.js, Express.js
- MongoDB database, Mongoose ODM

AI Integration: Hugging Face API (FLAN-T5 Model)

Development Tools:

- VS Code or any IDE
- Nodemon, dotenv, Git

Hardware Requirements

1. Processor: Intel i3 / AMD Ryzen 3 or higher
2. RAM: 8 GB or more
3. Storage: Minimum 100 MB free space for project files
4. Display: 1366 x 768 or higher resolution
5. Internet Connection: Required for AI API calls and frontend-backend communication

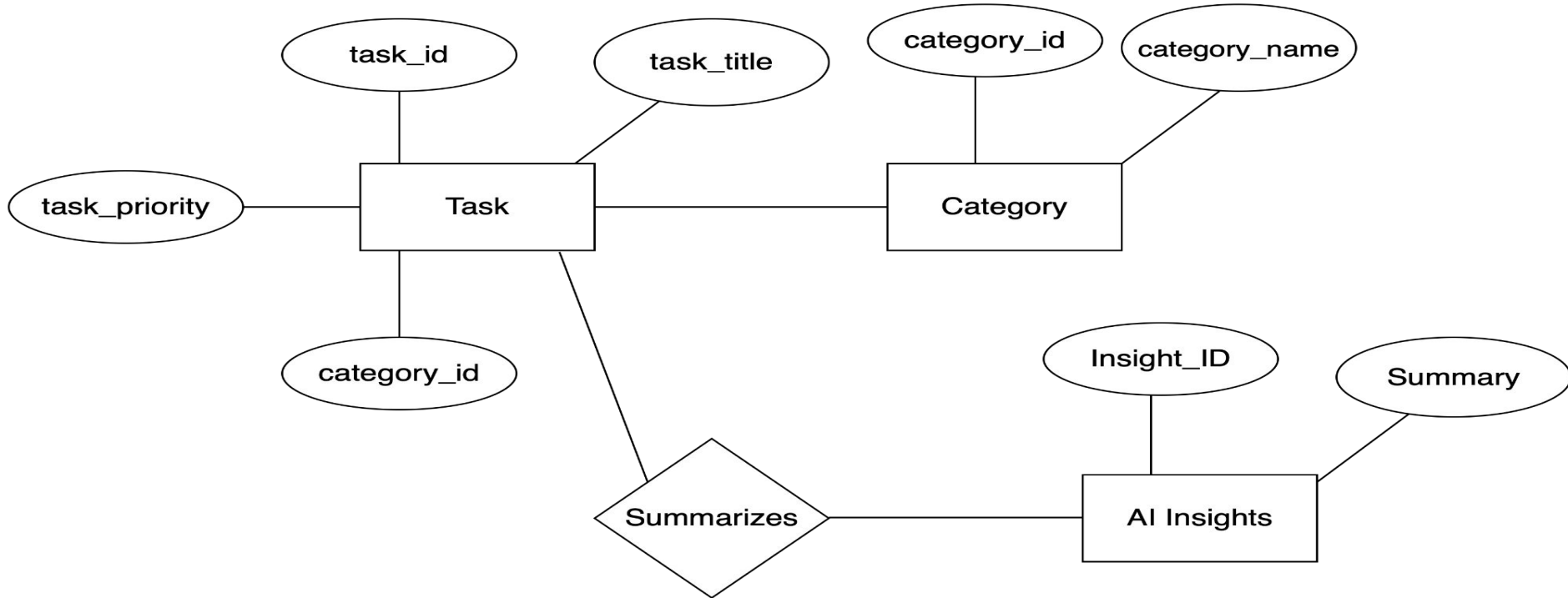


Data Structure and Concepts Used

- **Priority Queue (Heap):**
Used to manage tasks by priority (High > Medium > Low).
- **HashMap:**
Stores and retrieves tasks by category.
- **Queue:**
Handles sequential AI processing requests.
- **Stack:**
Used for undo/redo of task operations.
- **ArrayList:**
Stores all user tasks dynamically for frontend rendering.
- **Complexity:** $O(\log n)$ for insert/delete, $O(1)$ for retrieval.



ER diagram of the proposed system





Algorithm Explanation

- User enters a task description.
- Backend sends it to Hugging Face API for NLP analysis.
- AI returns priority, category, and estimated time.
- Task is added to Priority Queue based on priority.
- Sorted tasks displayed in React.js UI under relevant columns.
- On completion, task removed and stored in completed logs.
- At day's end, AI generates a summary report.



Time And Space Complexity

- **Insert (Heap):** $O(\log n)$
- **Delete:** $O(\log n)$
- **Retrieve Max Priority:** $O(1)$
- **AI Analysis:** $O(n)$ per input text
- **Overall Efficiency:** $O(n \log n)$ time, $O(n)$ space



Front End

Technology Used

- **React.js:** Modern JavaScript library for building interactive and responsive user interfaces.
- **CSS3:** For styling using Flexbox, Grid, and modern design features.
- **Axios:** For communicating with backend APIs and fetching data.
- **React DnD / Framer Motion:** Drag-and-drop functionality and smooth animations.

Key Features:

Interactive UI: User-friendly interface for adding, editing, and viewing tasks.

Three-Column Layout: Tasks displayed by priority (High, Medium, Low).

Real-Time Updates: Instant updates when tasks are added, deleted, or moved.

Responsive Design: Works on desktops, tablets, and mobile devices.

Dark Mode: Modern dark theme for better user experience.



Implementation

1. Priority Queue (Heap)

- Tasks are stored in a min-heap based on priority (1 = High, 2 = Medium, 3 = Low).
- Ensures highest priority tasks are always accessed first.
- Supports insert, delete, and retrieve operations efficiently.

2. AI Integration

- **Hugging Face FLAN-T5 Model** analyzes task descriptions to:
 - Assign priority
 - Detect category (Work, Personal, Health)
 - Predict completion time and deadline
- **Fallback mechanism:** keyword-based detection if AI fails.



Test Cases

Test ID	Description	Expected Output
TC01	Add a task	Task stored successfully
TC02	AI categorization	Returns correct category
TC03	Priority Queue order	High priority task first
TC04	Delete task	Task removed correctly
TC05	API failure	Fallback keyword logic works



Challenges And Solutions

Challenges	Solutions
API latency	Used caching for repeated inputs
AI misclassification	Added keyword fallback mechanism
Database performance	Implemented MongoDB indexing
UI lag	Used React.memo and useCallback



Future Scope

- Add voice assistant for voice-based task input.
- Integrate with Google Calendar / Outlook.
- Develop mobile app version (React Native).
- Introduce team collaboration features.
- Use machine learning for personalized task prediction.



Code

Priority Queue Implementation:

```
class PriorityQueue {  
  constructor() { this.heap = []; }  
  push(item, priority) {  
    this.heap.push({ item, priority });  
    this.heap.sort((a, b) => a.priority - b.priority);  
  }  
  pop() { return this.heap.shift(); }  
}
```

AI Integration (Hugging Face):

```
const response = await axios.post(  
  "https://api-inference.huggingface.co/models/flan-t5",  
  { inputs: taskDescription },  
  { headers: { Authorization: `Bearer ${API_KEY}` } }  
);
```



Output Screenshot

PriorityFlow

Intelligent task management with AI-powered prioritization and drag-and-drop organization

AI-Based Priority To-Do List

Tasks are automatically prioritized using AI or keyword analysis.
Drag and drop tasks between priority levels.

6

Total Tasks

6

Pending

0

Completed

Daily Summary

Enter a task (e.g., 'Finish report by Friday')

Add Task

Search tasks...

High Priority

Urgent DSA Submission Tomorrow

Uncategorized Oct 8, 2025

HIGH PRIORITY

Complete

Delete

Finish quarterly report by Friday deadline

Work

HIGH PRIORITY

Complete

Delete

Medium Priority

Schedule team meeting for next week

Work Oct 14, 2025

MEDIUM PRIORITY

Complete

Delete

Prepare presentation slides

Work

MEDIUM PRIORITY

Complete

Delete

Read book for personal development

Personal

MEDIUM PRIORITY

Complete

Delete

Low Priority

Video Games (TimePass)

Personal

LOW PRIORITY

Complete

Delete



Output Screenshot

Completed Tasks (3)

Delete All

Urgent DSA Submission Tomorrow

Uncategorized

HIGH PRIORITY

Undo

Delete

Schedule team meeting for next week

Work

MEDIUM PRIORITY

Undo

Delete

Video Games (TimePass)

Personal

LOW PRIORITY

Undo

Delete

Daily Summary

×

You completed 3 of 6 tasks today.

6

Total Tasks

3

Completed

50%

Completion

Task Categories

Uncategorized 1

Work 3

Personal 2



Conclusion

The AI-Based Priority To-Do List integrates Data Structure concepts (Priority Queue, HashMap, Queue) with AI intelligence for efficient task management.

Automatically prioritizes, categorizes, and estimates deadlines for tasks using AI models.

Provides a user-friendly interface with real-time updates, drag-and-drop, and a responsive design.

Key Achievements:

1. **Efficient Task Prioritization:** Highest priority tasks are always accessible using Heap/Priority Queue.
2. **AI-Powered Intelligence:** Smart classification, time estimation, and deadline prediction.
3. **Smart Summaries & Insights:** Daily summaries motivate and improve productivity.
4. **Seamless Full-Stack Integration:** React frontend, Node.js/Express backend, MongoDB storage.
5. **Scalable & Extensible:** Architecture allows addition of new AI features and optimizations.



References

- Hugging Face, “FLAN-T5 Model API Documentation,” 2024.
- React.js Official Docs, Meta Platforms Inc., 2024.
- Node.js Foundation, “Express Framework Documentation,” 2024.
- MongoDB Inc., “NoSQL Database Developer Guide,” 2024.
- McMillan, M., *Data Structures and Algorithm Analysis in JavaScript*, 2023.