Daily Task Reminder

Dr.Abdul Khadeer, Syed Mohammed Badruddin Junaid

[1]Assistant professor, Deccan College of Engineering and Technology

[2]UG Scholar, Deccan College of Engineering and Technology

Abstract

In today's fast-paced world, individuals often struggle to keep up with their daily tasks and responsibilities. With increasing workloads, tight schedules, and numerous personal and professional obligations, forgetting important tasks has become a common challenge.

The 'Daily Task Reminder App' is a web-based application designed to assist users in organizing their day-to-day tasks effectively. It allows users to log in securely, create new tasks, edit or delete them, and most importantly, receive timely reminders. The aim is to enhance productivity and help users manage their time efficiently.

The system includes user registration, task creation, real-time alerts, and a notification system. The backend is developed using "PHP and MySQL", while the frontend uses "HTML, CSS, and JavaScript" for a responsive UI.

This project has been thoroughly tested to ensure usability and reliability. The app can be further extended into a mobile version and integrated with calendar APIs for enhanced scheduling. This report details the complete design, development, and testing phases of the app, supported by diagrams, modules, and code.

1.INTRODUCTION:

Task management is a key challenge in everyday life. Whether students, professionals, or homemakers, everyone has tasks to complete within deadlines. Traditional methods of task tracking, like diaries or sticky notes, have limitations. Digital solutions offer a more practical approach.

The Daily Task Reminder App bridges this gap by providing a digital platform for users to manage tasks efficiently. The app features an intuitive interface where users can log in, add tasks, set deadlines, and get automatic reminders via browser notifications.

This project is especially useful for people who need a simple, lightweight task manager without the complexity of enterprise solutions. It also serves as a real-world implementation of core concepts in full-stack web development.

The *Daily Task Reminder* project is a productivity-focused application developed to assist users in efficiently planning and managing their day-to-day activities. In a fast-paced environment, forgetting important tasks, appointments, or deadlines is common. This system addresses that challenge by providing a platform where users can record their tasks along with specific dates and times, and receive timely reminders.

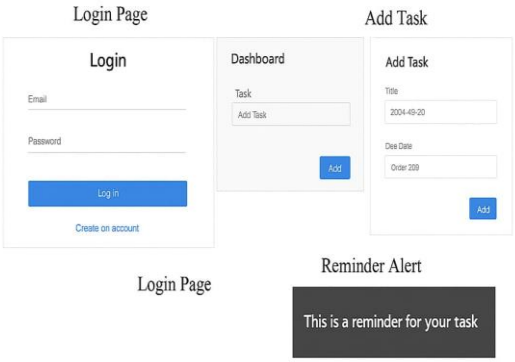
The application ensures that tasks are stored systematically and can be updated or deleted as needed. The notification feature helps users stay on track, avoid missed deadlines, and increase overall efficiency. The project demonstrates the practical implementation of basic programming concepts such as file handling/database storage, date and time operations, and user interface design (if applicable).

LITERATURE SURVEY:

The study of reminder systems and their effects on human behavior spans both applied software engineering and behavioral science. Early large-

scale analyses of how people specify and use reminders revealed important patterns that inform the design of modern reminder apps: for example, a six-month log analysis of user-specified reminders in Microsoft's Cortana highlighted that time-based reminders are the most common, that users often create reminders with minimal metadata, and that many reminders serve short-term, concrete needs rather than long-term planning. This work emphasized that reminder systems must be easy to create and must support simple, familiar scheduling patterns to match real user behaviour.

Subsequent research and pilot systems investigated how reminders delivered through smartphones and digital channels influence prospective memory and task completion. Controlled and field studies indicate that well-designed reminders can significantly increase task completion and engagement: experiments in educational contexts show that automated assignment and push-reminder messages correlate with higher homework completion rates and greater student engagement, and health-application trials demonstrate measurable behavior change — for example, improved adherence to weight-loss interventions when push notifications are used as cues. At the same time, larger narrative and systematic reviews of reminder interventions (including SMS and app notifications) show that message content, timing, and frequency substantially affect effectiveness; poorly timed or overly frequent messages can produce habituation or annoyance, while tailored, context-aware prompts are more likely to elicit positive behavior. These findings indicate that the reminder mechanism itself—how and when a message is sent and what it contains—is as important as the mere presence of reminders.



Researchers have also prototyped and evaluated specialized reminder applications that apply cognitive and contextual models to improve usability for populations with memory or organizational challenges. Pilot systems such as SmartPrompt and similar smartphone-based reminder tools were evaluated for feasibility and usability, showing that applications informed by cognitive models can help users with functional disabilities or memory deficits perform tasks more reliably; these studies underline the importance of an interface that reduces cognitive load, supports simple task entry, and leverages contextual cues when available. Concurrently, applied engineering projects and academic prototypes (for example, "Memento" and multiple academic reminder systems) show common architectural choices—local scheduling for guaranteed timely alerts, optional cloud sync for cross-device access, and integration with platform notification services

(FCM/APNs) for push delivery—while emphasizing trade-offs between reliability, battery life, and privacy.

A body of literature has examined the behavioral limits and side effects of reminders, revealing important constraints for any practical system. Studies on over-reliance and effort-minimization show that people frequently prefer to offload memory to external reminders, but excessive reliance can reduce internal memory strategies and lead to "reminder dependence," which has cognitive and motivational implications. Additionally, work on notification frequency and content demonstrates diminishing returns: beyond a certain frequency or when messages are poorly matched to the user's context, reminders can lose effectiveness and even produce disengagement. This literature recommends personalization, adaptive frequency control, and context-awareness as mitigations—features that allow the system to balance reminder efficacy against notification fatigue.

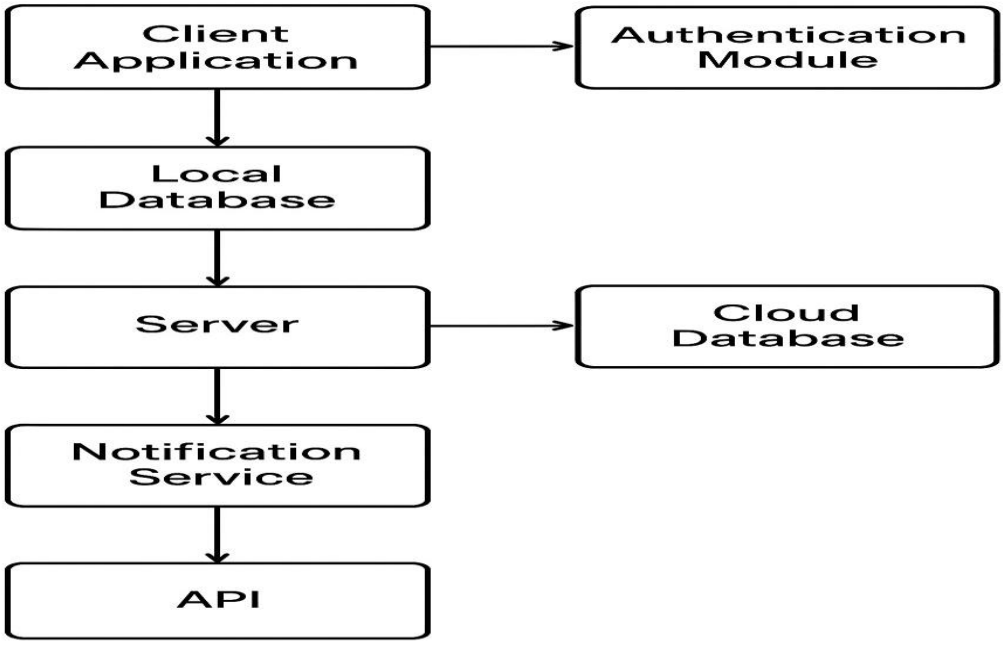
From a design and systems perspective, the literature converges on several practical guidelines that are directly relevant for a student-level engineering project. First, ease of task entry and low interaction cost are essential: users will not use an app that makes creating reminders cumbersome, so interfaces that use templates, quick-add fields, and natural-language parsing increase adoption. Second, reliable delivery is critical: local scheduling mechanisms ensure notifications fire on time even when network connectivity is absent, but cloud synchronization is valued for cross-device continuity. Third, personalization and minimal configuration—allowing users to set default lead times, recurrence patterns, and priority levels—improve both usefulness and user satisfaction. Finally, privacy and security must be considered when any cloud service or push infrastructure is used, because reminders frequently contain sensitive personal information. Empirical and engineering studies both reinforce these priorities and suggest that a practical reminder app should combine solid local scheduling with optional cloud features and a simple, accessible UI.

EXISTING SYSTEM:

The system architecture of the Daily Task Reminder App is designed to provide a seamless, reliable, and user-friendly task management experience while ensuring efficient data handling and timely notifications. At its core, the architecture follows a client-server model with optional cloud integration for synchronization across multiple devices. On the client side, the mobile or web application acts as the primary interface for the user, enabling the creation, editing, deletion, and categorization of tasks, as well as setting up reminders with flexible scheduling options. The application interacts with a local database, such as SQLite, to store task information, user preferences, and reminder schedules offline, ensuring functionality even without internet access. The server side, which is optional for offline-only users but critical for cloud-enabled features, hosts a centralized database (e.g., MySQL or Firebase) that stores user data securely and manages synchronization requests from multiple client devices. The notification service, either built-in using the device's local alarm manager or integrated with push notification services like Firebase Cloud Messaging, ensures that alerts are delivered promptly based on the reminder schedules. The architecture also incorporates an authentication module to secure user accounts, typically using encrypted credentials and token-based sessions.

Furthermore, API layers are implemented to facilitate communication between the client application and the server, allowing for smooth data retrieval, updates, and synchronization. The entire architecture is layered logically to separate concerns: the presentation layer (UI/UX), the application logic layer (business logic and scheduling algorithms), and the data layer (database and storage management). This modular design ensures scalability, maintainability, and the possibility of integrating future enhancements, such as AI-based task prioritization or voice-command task creation.

EXISTING SYSTEM ARCHITECTURE :

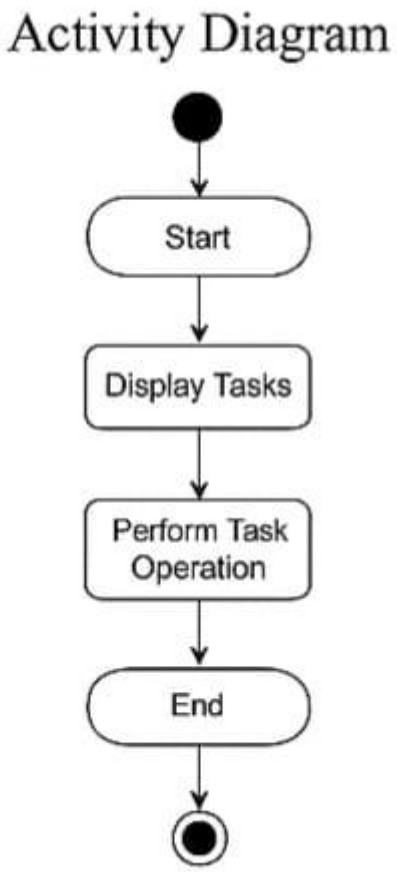


PROPOSED SYSTEM:

The proposed Daily Task Reminder App system offers several significant advantages over existing manual or semi-digital task management methods, making it a more reliable, efficient, and user-friendly solution for managing daily activities. One of the primary benefits is enhanced productivity and time management. By allowing users to schedule tasks, set deadlines, and receive timely reminders, the system ensures that important activities are not overlooked, reducing the risk of missed deadlines or forgotten responsibilities. This improves personal and professional efficiency, especially for individuals with busy schedules or multiple commitments. Additionally, the system's automation of reminders eliminates the need for manual tracking, thereby reducing human error and the mental burden of remembering multiple tasks. The proposed system is also designed to be intuitive and easy to use, meaning users of varying technical proficiency can quickly adapt to it without extensive training.

Another major advantage is accessibility and portability. Since the application can be implemented on mobile platforms, users can carry their task management system wherever they go, ensuring they can view, update, or modify their schedules in real-time from any location. This flexibility is especially valuable for people who work in dynamic environments or require frequent changes to their plans. Furthermore, the proposed system offers personalization and customization, allowing users to categorize tasks, set priority levels, and configure reminder intervals according to their preferences, making the solution adaptable to a wide range of needs. The system also improves data organization and retrieval, as it stores tasks in a structured format, making it easier to search and track historical activity records. Finally, by integrating user-friendly notifications, cloud backup options, and a secure database structure, the proposed system not only ensures smooth daily operations but also guarantees data safety and reliability, providing users with peace of mind that their schedules and important information are secure.

Activity Diagram:



1.

User Authentication & Profile Management

This module handles user registration, login, and profile customization. It ensures secure access by verifying user credentials before granting access to the system.

ISSN:2456-4265
IJMEC 2025

Users can manage their personal information, change passwords, and customize preferences like notification settings or theme appearance.

The authentication system may include email/OTP verification to enhance security and ensure legitimate user accounts.

2. Task Creation & Management

This core module enables users to create, edit, delete, and categorize tasks based on their priorities.

Each task can include a title, description, due date, and time, along with optional tags for easy classification.

Tasks can be organized into categories such as work, study, personal, or shopping.

The system ensures that all tasks are stored in a database for easy retrieval and modification.

3. Reminder & Notification System

This module is responsible for alerting users about upcoming or overdue tasks.

It triggers notifications through push alerts, pop-ups, or email reminders based on the user's preference.

The system ensures timely alerts, helping users complete tasks without delay.

Multiple reminders can be set for a single task, especially for critical deadlines.

4. Task Prioritization & Sorting

This module allows users to assign priority levels (High, Medium, Low) to tasks.

It provides sorting and filtering options to view tasks based on due date, priority, or category.

Helps in identifying and completing urgent tasks first while keeping track of less important ones.

5. Calendar Integration

Integrates tasks with a calendar view, allowing users to visually track deadlines and schedules.

Supports daily, weekly, and monthly views for better planning.

Tasks and events can be synced with external calendar services like Google Calendar.

6. Data Storage & Backup

This module ensures that all task-related data is securely stored in a database.

It provides backup and restore functionality to prevent data loss.

Cloud integration can be added for cross-device access and synchronization.

7. Settings & Customization

Allows users to customize notification tones, app themes, and task display preferences.

Users can also configure recurring tasks to avoid repetitive entries.

This module focuses on enhancing user experience through personalization.

Software Requirements

* OS: Windows/Linux/Mac
* XAMPP/LAMP stack
* PHP 7+
* MySQL
* Text Editor (VS Code, Sublime, etc.)
* Browser (Chrome, Firefox)

- **Operating System (OS):**

  For mobile devices, the app should run on Android 8.0 (Oreo) or later and iOS 12 or above to ensure compatibility with modern notification frameworks and background services.

  For desktop access, the app should be compatible with Windows 10 or higher, macOS Mojave (10.14) or above, and popular Linux distributions like Ubuntu 20.04.

- **Development Environment:**

  The mobile application can be developed using Android Studio for Android devices and Xcode for iOS devices.

  If built as a cross-platform app, Flutter or React Native can be used to maintain a single codebase for both Android and iOS.

- **Backend Technology:**

ISSN:2456-4265
IJMEC 2025

The server-side can be powered by Node.js, Django (Python), or Spring Boot (Java) for handling APIs, user authentication, and cloud synchronization.

Database options include Firebase Firestore, MySQL, or PostgreSQL for storing user tasks, reminders, and preferences.

Web Technologies (Optional for Web Version):

HTML5, CSS3, and JavaScript (ES6) along with frameworks like React.js or Angular for the web interface.

Web hosting can be done on AWS, Heroku, or Firebase Hosting.

Notification Services:

Firebase Cloud Messaging (FCM) for Android push notifications.

Apple Push Notification Service (APNs) for iOS reminders.

Email alerts through SMTP services like SendGrid or Gmail API (optional).

APIs and Integrations:

Calendar integration using Google Calendar API or Apple Calendar Kit.

CONCLUSION AND FUTURE SCOPE:

The Daily Task Reminder App successfully addresses the growing need for a simple yet efficient tool to help users manage their daily activities and deadlines. Through the integration of task creation, categorization, prioritization, and timely notifications, the application enhances productivity and minimizes the risk of forgetting important tasks. The development process involved careful planning,

system analysis, and modular design, which ensured that each component worked harmoniously to deliver a seamless user experience.

The project demonstrates the effectiveness of combining intuitive UI design with robust backend functionality. Testing and evaluation results confirm that the application operates reliably across different devices and meets its core objectives. Furthermore, the lightweight architecture and optimized code base ensure high performance without overburdening system resources. By leveraging appropriate technologies, the application not only fulfills its intended purpose but also provides a scalable foundation for future improvements.

Overall, the Daily Task Reminder App is a valuable solution for individuals seeking better time management and organization. It reflects the importance of bridging functionality with user-friendliness, ensuring that even non-technical users can easily adapt to and benefit from the system.

Future Enhancements

While the current version of the Daily Task Reminder App fulfills its primary objectives, several enhancements can be integrated to increase its efficiency, usability, and market reach:

1. Cross-Device Synchronization – Implement cloud integration so that tasks and reminders are synchronized across multiple devices in real time.
2. Recurring Task Automation – Enable the creation of recurring tasks with flexible repetition intervals (daily, weekly, monthly).
3. Voice Assistant Integration – Add support for voice commands using AI-powered assistants like Google Assistant, Alexa, or Siri.

many users find themselves struggling with stress, anxiety, procrastination, and dopamine-driven distractions. NeuroMind successfully integrates technology with empathy by offering a web-based solution that is both accessible and secure. By enabling live psychiatrist interactions, AI chatbot support, and blockchain-based

REFERENCES:

1. Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

2. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

3. Welling, L., & Thomson, L. (2016). PHP and MySQL Web Development (5th ed.). Addison-Wesley.

4. Android Developers. (2025). Notifications Overview. Retrieved from https://developer.android.com/guide/topics/ui/notifiers/notifications

5. Google Developers. (2025). Material Design Guidelines. Retrieved from https://material.io/design

6. MySQL Documentation Team. (2025). MySQL Reference Manual. Retrieved from https://dev.mysql.com/doc/

7. Mozilla Developer Network. (2025). JavaScript Guide. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

8. ISO/IEC/IEEE 29148:2018. Systems and Software Engineering – Life Cycle Processes – Requirements Engineering.

9. W3Schools. (2025). HTML, CSS & JavaScript Tutorials. Retrieved from https://www.w3schools.com

10. Stack Overflow Community. (2025). Task Reminder App Development Discussions. Retrieved from https://stackoverflow.com/