

# Task Manager App Documentation

## Overview

Task Manager is a modern web application built with React and Firebase that allows users to create, manage, and track their tasks. The application features user authentication, task filtering, search capabilities, deadline notifications, and theme customization.

## Features

- **User Authentication:** Secure login and registration using Firebase Authentication
- **Task Management:** Create, edit, delete, and mark tasks as complete
- **Task Prioritization:** Assign priority levels to tasks (low, medium, high)
- **Deadline Management:** Set deadlines for tasks with approaching deadline notifications
- **Search & Filtering:** Find tasks quickly with search functionality and filter by status
- **Theme Switching:** Toggle between light and dark modes for comfortable viewing
- **Responsive Design:** Optimized for both desktop and mobile devices

## Technical Architecture

### Frontend Technologies

- **React:** UI library for building the user interface
- **Firebase Authentication:** For user authentication
- **Firestore:** Real-time database for task storage
- **CSS:** Custom styling with responsive design

### Project Structure

```
TO-DO/
├── src/
│   ├── assets/           # Static assets
│   ├── components/       # React components
│   ├── firebase/         # Firebase configuration
│   ├── App.jsx           # Main application component
│   ├── App.css           # Main application styles
│   ├── index.css         # Global styles and CSS variables
│   └── main.jsx          # Application entry point
├── public/
└── index.html            # HTML template
```

— package.json	# Project dependencies
— vite.config.js	# Vite configuration
— eslint.config.js	# ESLint configuration

## Core Components

### App.jsx

The main component that orchestrates the application. It:

- Manages application state
- Handles user authentication
- Connects to Firestore database
- Provides task management functions
- Controls theme switching
- Renders the appropriate UI based on authentication status

### Firebase Integration

The application uses Firebase for:

1. **Authentication:** Login, registration, and user management
2. **Database:** Storing and retrieving tasks with real-time updates
3. **Security:** Ensuring users can only access their own tasks

## User Guide

### Getting Started

1. **Create an Account:** Register with email or sign in with Google
2. **Add a Task:** Enter task details in the input field and click the add button
3. **Manage Tasks:** Mark tasks as complete, edit, or delete them as needed

### Task Management

- **Adding Tasks:** Enter task text in the input field and click the add button
- **Setting Priority:** Select priority level (low, medium, high) when creating or editing a task
- **Setting Deadlines:** Add optional deadlines to tasks for time management
- **Editing Tasks:** Click the edit icon to modify task details
- **Completing Tasks:** Click the checkbox to mark a task as complete
- **Deleting Tasks:** Click the delete icon to remove a task

## Filtering and Searching

- **Filter Options:** View all tasks, active tasks only, or completed tasks
- **Search:** Use the search bar to find specific tasks by text
- **Clear Completed:** Remove all completed tasks at once

## Theme Customization

- **Theme Toggle:** Switch between light and dark modes using the theme toggle button
- **Persistence:** Your theme preference is saved between sessions

# Developer Guide

## Setup Instructions

### 1. Prerequisites:

- Node.js (latest LTS version recommended)
- npm(recommended) or yarn

### Installation:

```
# Clone the repository  
git clone [repository-url]
```

```
# Navigate to project directory  
cd task-manager
```

```
# Install dependencies  
npm install
```

```
# Run development server  
npm run dev
```

2.

### 3. Firebase Setup:

- Create a Firebase project at [firebase.google.com](https://firebase.google.com)
- Enable Authentication (Email/Password and Google provider)
- Create a Firestore database
- Update the Firebase configuration in `src/firebase/config.js`

## Key Files

- **App.jsx**: Main application logic and state management
- **firebase/config.js**: Firebase configuration and service initialization
- **components/**: Individual UI components for task management
- **App.css & index.css**: Application styling

## Data Structure

Firestore collection structure:

- **tasks**:
  - **text**: String (task description)
  - **completed**: Boolean (completion status)
  - **priority**: String (low, medium, high)
  - **createdAt**: Timestamp (when the task was created)
  - **deadline**: String (optional deadline date)
  - **notificationSent**: Boolean (whether notification has been sent)
  - **userId**: String (reference to user)

## Advanced Features

### Deadline Notifications

The application checks for approaching deadlines and sends notifications:

1. Tasks with deadlines within one hour trigger notifications
2. Notification status is tracked to prevent duplicate alerts
3. The application periodically checks for upcoming deadlines

### Authentication Flow

1. **Registration**: New users can create accounts with email/password
2. **Login**: Existing users can log in with their credentials
3. **Google Authentication**: Users can sign in with their Google accounts
4. **Persistent Login**: Authentication state persists between sessions

# Future Enhancements

Potential improvements for future versions:

1. **Task Categories:** Organize tasks by custom categories or projects
2. **Recurring Tasks:** Set up repeating tasks on daily, weekly, or monthly schedules
3. **Task Analytics:** View statistics about task completion and productivity
4. **Collaborative Tasks:** Share tasks with other users for team collaboration
5. **Mobile App:** Native mobile application with push notifications
6. **Enhanced Notifications:** Email and SMS notifications for important deadlines
7. **Data Export:** Export task data in various formats (CSV, PDF)