



Digital Egypt Pioneers Initiative

React Front-End Development

Taskfyer – Task Management System

Supervised by:

Eng. Basma Abdel Halim

Author:

Ahmed Nasser Abdelfadil

Acknowledgement

I would like to extend my sincere gratitude to Eng. Basma Abdel Halim for her continuous support and invaluable guidance throughout the development of my Taskfyer – Task Management System project, completed as part of the Digital Egypt Pioneers Initiative. Her mentorship has been instrumental in expanding my knowledge and sharpening my skills, allowing me to approach challenges with greater clarity and confidence.

Her expertise, constructive feedback, and commitment to fostering innovation significantly contributed to the refinement of my project. Through her thoughtful advice and practical suggestions, I was able to navigate technical and conceptual hurdles with greater ease.

I am truly grateful for the opportunity to learn and grow under her supervision. Her dedication and encouragement have had a lasting impact on my development, and I deeply appreciate her unwavering support throughout this journey.

Abstract

Taskfyer is a task management system designed to help individuals and teams effectively organize, track, and manage their daily responsibilities. The system aims to streamline workflow by offering features such as task creation, editing, prioritization, status tracking, and deadline management through a user-friendly interface.

Taskfyer focuses on simplicity, usability, and productivity. The application supports task categorization, real-time updates, and intuitive interactions to enhance user experience and efficiency. By leveraging modern web development technologies, the system ensures responsive design and smooth performance across devices.

Taskfyer not only addresses the challenges of disorganization and poor time management but also promotes goal setting and accountability. It serves as a practical tool for students, professionals, and teams seeking to improve their task management habits in a digital-first environment.

Introduction

Taskfyer is a full-stack web application designed to streamline task management for individuals and teams. It offers a user-friendly interface, secure authentication, and robust task management features, making it an ideal tool for students, professionals, and collaborative teams. By addressing common issues like disorganization and missed deadlines, Taskfyer promotes productivity and accountability. This documentation provides a comprehensive overview of the project, including its features, architecture, setup instructions, and usage guidelines.

Motivation

The motivation behind Taskfyer stems from the widespread need for efficient task management solutions. Many individuals and teams struggle with disorganized workflows, missed deadlines, and lack of clear task prioritization. Taskfyer was created to provide a simple yet powerful tool that helps users stay on top of their responsibilities, offering features like role-based access control and intuitive status tracking to cater to diverse user needs.

Features

Taskfyer offers a range of features to enhance task management:

- **User Authentication:** Secure registration, login, email verification, and password reset.
- **Role-Based Access Control:** Different permissions for admins (manage users), creators (create and assign tasks), and users (manage personal tasks).
- **Task Management:** Create, read, update, and delete tasks with options to set titles, descriptions, statuses (e.g., todo, in progress, done), due dates, and assignees.
- **Admin Panel:** Tools for administrators to view and delete user accounts.
- **Responsive Design:** A seamless experience across desktops, tablets, and smartphones.
- **Email Notifications:** Automated emails for verification and password reset using Nodemailer.

System Architecture

Taskfyer follows a client-server architecture:

- **Frontend:** Built with Next.js and React, providing a dynamic and responsive user interface.
- **Backend:** Developed using Node.js and Express.js, handling API requests and business logic.
- **Database:** MongoDB, a NoSQL database, stores user and task data, managed via Mongoose.
- **Email Services:** Integrated with Nodemailer for sending verification and password reset emails.

The frontend communicates with the backend through RESTful API endpoints, ensuring secure and efficient data exchange.

Technologies Used

Backend

Technology	Purpose
Node.js	Server-side JavaScript runtime
Express.js	Web framework for RESTful APIs
MongoDB	NoSQL database
Mongoose	MongoDB ODM
Nodemailer	Email communication
JWT	User authentication
dotenv	Environment variable management

Frontend

Technology	Purpose
Next.js	Server-side rendering and routing
React	Component-based UI rendering
Tailwind CSS	Utility-first CSS framework
react-hot-toast	Notification system
React Context API	State management

Database Schema

Taskfyer uses two main MongoDB collections:

Users Collection

Field	Type	Description
_id	ObjectId	Primary key
name	String	Full name of user
email	String	User email address
password	String	Encrypted password
role	String	admin / creator / user
verified	Boolean	Email verification status
verificationToken	String	Token used for verification
resetPasswordToken	String	Token for password reset
resetPasswordExpire	Date	Expiration date for reset token

Tasks Collection

Field	Type	Description
_id	ObjectId	Unique ID
title	String	Task title
description	String	Task description
status	String	todo / in progress / done
createdBy	ObjectId	User who created the task
assignedTo	ObjectId	Assigned user
dueDate	Date	Task deadline
createdAt	Date	Creation timestamp
updatedAt	Date	Last update timestamp

API Documentation

The backend provides a RESTful API with key endpoints:

Method	Endpoint	Description
POST	/api/v1/register	Register new user
POST	/api/v1/login	User login
GET	/api/v1/verify/:token	Email verification
POST	/api/v1/forgotpassword	Request password reset
PUT	/api/v1/resetpassword/:token	Reset password
GET	/api/v1/admin/users	View all users (Admin only)
DELETE	/api/v1/admin/user/:id	Delete user (Admin only)
POST	/api/v1/task/create	Create new task
GET	/api/v1/task/:id	View task by ID
PUT	/api/v1/task/update/:id	Update task details
DELETE	/api/v1/task/delete/:id	Delete a task
GET	/api/v1/tasks	Get tasks for logged-in user

For detailed endpoint usage, refer to the backend repository.

Setup and Installation

Prerequisites

- Node.js (version 16 or higher)
- MongoDB (local or cloud-based, e.g., MongoDB Atlas)
- Gmail account with an App Password

Backend Setup

1. Clone the backend repository:

```
git clone https://github.com/ahmednasser111/task-management-system-api.git
```

2. Navigate to the backend directory:

```
cd task-management-system-api
```

3. Install dependencies:

```
npm install
```

4. Create a .env file with:

5. MONGO_URI=your_mongodb_connection_string
6. JWT_SECRET=your_jwt_secret_key
7. PORT=8000
8. CLIENT_URL=https://task-management-system-tawny-ten.vercel.app
9. USER_EMAIL=your_gmail_address
10. EMAIL_PASS=your_gmail_app_password
- NODE_ENV=development

11. Start the server:

```
npm run dev
```

The server runs at <http://localhost:8000>.

Frontend Setup

1. Clone the frontend repository:

```
git clone https://github.com/ahmednasser111/task-management-system.git
```

2. Navigate to the frontend directory:

```
cd task-management-system
```

3. Install dependencies:

```
npm install
```

4. Create a .env.local file with:

```
NEXT_PUBLIC_SERVER_URL=https://task-management-system-api-  
production.up.railway.app
```

5. Start the development server:

```
npm run dev
```

The frontend runs at <http://localhost:3000>.

Note: Ensure the backend server is running before starting the frontend.

Usage Guide

User Registration and Login

1. Visit Taskfyer to register.
2. Enter your name, email, and password.
3. Verify your email via the link sent to your inbox.
4. Log in at Taskfyer Login.

Task Management

- **Create a Task:** Navigate to the task creation page, enter title, description, status, due date, and assignee (if applicable).
- **View Tasks:** Access the task list to see all tasks with their statuses and deadlines.
- **Update Tasks:** Edit task details or change status.
- **Delete Tasks:** Remove unnecessary tasks.

Admin Functionalities

Admins can access the admin panel to:

- View all registered users.
- Delete user accounts as needed.

Deployment

Taskfyer is deployed on:

- **Backend:** Vercel
- **Frontend:** Vercel

Deployment involves setting up environment variables on the respective platforms and using their CI/CD pipelines for automatic updates.

Challenges and Solutions

- Implemented JWT and role-based access control
- Used Gmail App Password for Nodemailer
- Debugged deployment via Railway/Vercel logs
- Switched email methods due to SMTP restrictions

Links:

- Back-end github repo: <https://github.com/ahmednasser111/task-management-system-api>
- Front-end github repo: <https://github.com/ahmednasser111/task-management-system>
- Front-end live: <https://task-management-system-tawny-ten.vercel.app/>
- Back-end live: <https://task-management-system-api-bay.vercel.app/>