Note Management Application Documentation

1 Introduction

This **Note Management Application** is designed to streamline personal and professional organization. Users can create notes, group them into folders, assign statuses, and set deadlines, making it easy to track tasks, thoughts, and goals with a structured, intuitive UI.

2 Application Architecture

This application follows a **monolithic architecture**, combining frontend and backend within the Laravel framework. It is **component-driven**, with custom hooks for data management and Inertia.js for linking React components with Laravel.

```
timeline/
├── app/
│   ├── Http/
│   │   ├── Controllers/
│   │   │   ├── Auth/
│   │   │   ├── Controller.php
│   │   │   ├── DocumentationController.php
│   │   │   ├── FolderController.php
│   │   │   ├── NoteController.php
│   │   │   └── ProfileController.php
│   ├── Models/
│   │   ├── Folder.php
│   │   ├── Note.php
│   │   └── User.php
├── resources/
│   ├── js/
│   │   ├── Components/
│   │   │   ├── Forms/
│   │   │   └── Wrappers/
│   │   ├── Hooks/
│   │   ├── Layouts/
│   │   └── Pages/
│   │   ├── Auth/
│   │   ├── Profile/
│   │   ├── Dashboard.jsx
│   │   ├── Documentation.jsx
│   │   └── Notes.jsx
├── routes/
│   ├── api.php
│   ├── auth.php
│   ├── channels.php
│   ├── console.php
│   └── web.php
```

3 Features Overview

- **User Authentication**: Secure login, registration, and logout.
- **Note Management**: Create, edit, and delete notes with content fields.
- **Folder Organization**: Group notes within folders for easy categorization.
- **Status Assignment**: Assign statuses to notes to track progress.
- **Deadline Tracking**: Add start and end dates for task reminders.
- **Advanced Filtering**: Filter notes by status.

4 Installation

Follow these steps to set up the application locally:

1. Clone the repository: `$ git clone https://github.com/Fadi-N/timeline.git`
2. Install backend dependencies: `$ composer install`
3. Install frontend dependencies: `$ npm install`
4. Environment setup: Copy `$ .env.example` to `$ .env` and configure settings.
5. Database setup: Run migrations with `$ php artisan migrate`
6. Build the application: Compile assets with `$ npm run dev` and start the server with `$ php artisan serve`

5 Configuration

Variable	Description
DB_DATABASE	Name of the database
DB_USERNAME	Database user
DB_PASSWORD	Password for database user

6 Usage Guide

- **Homepage**: Introduction to the application with login/register options.
- **Dashboard**: Displays all folders created by the user.

- **Folder View**: Shows notes within a selected folder with status filtering.
- **Note Management**: Allows creation, editing, and deletion of notes.

7 Key Components

This application consists of several key components:

- **Controllers**: Manage HTTP requests and contain business logic.
- **Models**: Represent database entities and handle interactions.
- **Resources**: Frontend React components, hooks, and layouts.
- **Routes**: Define URL mappings to controllers and endpoints.

8 API Endpoints

Below is a list of the main API endpoints:

Authentication

- **POST /login**: Logs in a user and returns an authentication token.
- **POST /register**: Registers a new user and returns an authentication token.
- **POST /logout**: Logs out the authenticated user.

Folders

- **GET /folders**: Retrieves a list of folders.
- **POST /folders**: Creates a new folder.
- **PUT /folders/{id}**: Updates a folder by ID.
- **DELETE /folders/{id}**: Deletes a folder by ID.

Notes

- **GET /folders/{folder_id}/notes**: Retrieves notes in a folder.
- **POST /folders/{folder_id}/notes**: Creates a new note.
- **PUT /notes/{id}**: Updates a note by ID.
- **DELETE /notes/{id}**: Deletes a note by ID.

9 Security

This application employs several security measures to protect user data and application integrity:

- **Database Security**: Implements secure access controls, protecting data against unauthorized access. SQL Injection protection is enforced by using parameterized queries.

- **Encryption**: Sensitive data such as user passwords are encrypted using secure hashing algorithms to prevent exposure in the event of a data breach.
- **Input Validation**: Validates and sanitizes all user inputs to prevent common security vulnerabilities like XSS and SQL Injection attacks.

10 Authentication Mechanisms

The application uses secure authentication mechanisms to ensure only authorized users can access their data:

- **JWT Authentication**: Users are authenticated via JSON Web Tokens (JWT), ensuring secure and stateless sessions.
- **Password Hashing**: Passwords are hashed using a robust hashing algorithm, such as bcrypt, to safeguard user credentials.
- **Session Timeout**: Sessions automatically expire after a period of inactivity, adding another layer of protection against unauthorized access.

11 Deployment

The application can be deployed on various web servers. Here is a brief overview of the steps to deploy this application:

1. **Setup the Server**: Configure a web server environment (e.g., Apache, Nginx) with PHP and a MySQL database.
2. **Clone the Repository**: Use `$ git clone` to copy the application's repository to the server.
3. **Install Dependencies**: Run `$ composer install` and `$ npm install` to install backend and frontend dependencies.
4. **Environment Configuration**: Set up the `$ .env` file with production database and server credentials.
5. **Run Migrations**: Apply database migrations by running `$ php artisan migrate`.
6. **Start the Server**: Configure the web server to serve the application, and ensure assets are compiled with `$ npm run production`.

The application is live at: http://timeline-fn.cba.pl/

12 Troubleshooting

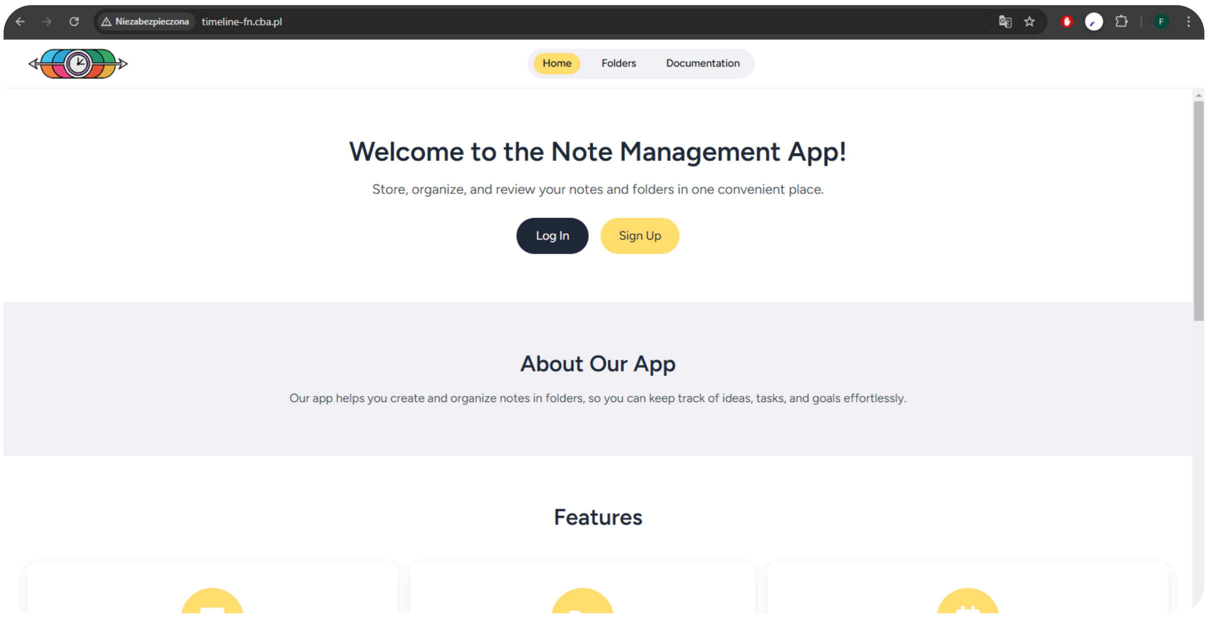
Issue: Database connection error
Solution: Ensure that the database credentials in the `$ .env` file are correct,

and that the database server is running. Check values for `$ DB_DATABASE`, `$ DB_USERNAME`, and `$ DB_PASSWORD`.

Issue: CSS or JavaScript not loading correctly
Solution: Run `$ npm run dev` to compile assets. If in production, run `$ npm run production` for minified assets.

Issue: "Class not found" error
Solution: Ensure all dependencies are installed by running `$ composer install` and `$ npm install`. If the issue persists, run `$ composer dump-autoload`.

13 Screenshots



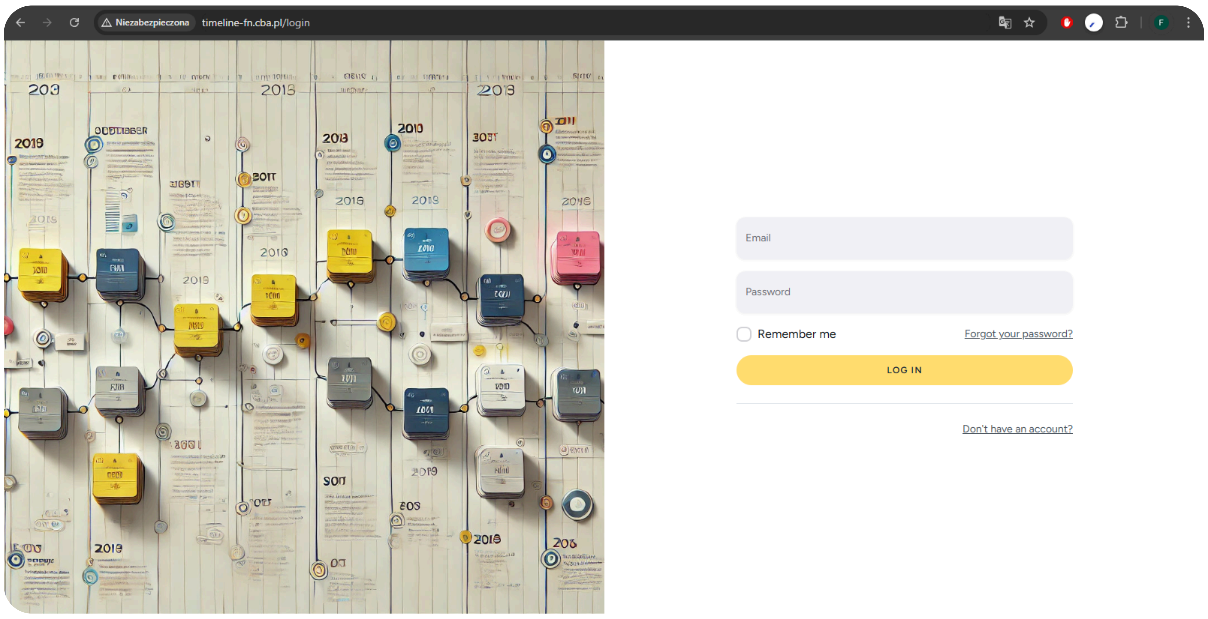
Welcome to the Note Management App!

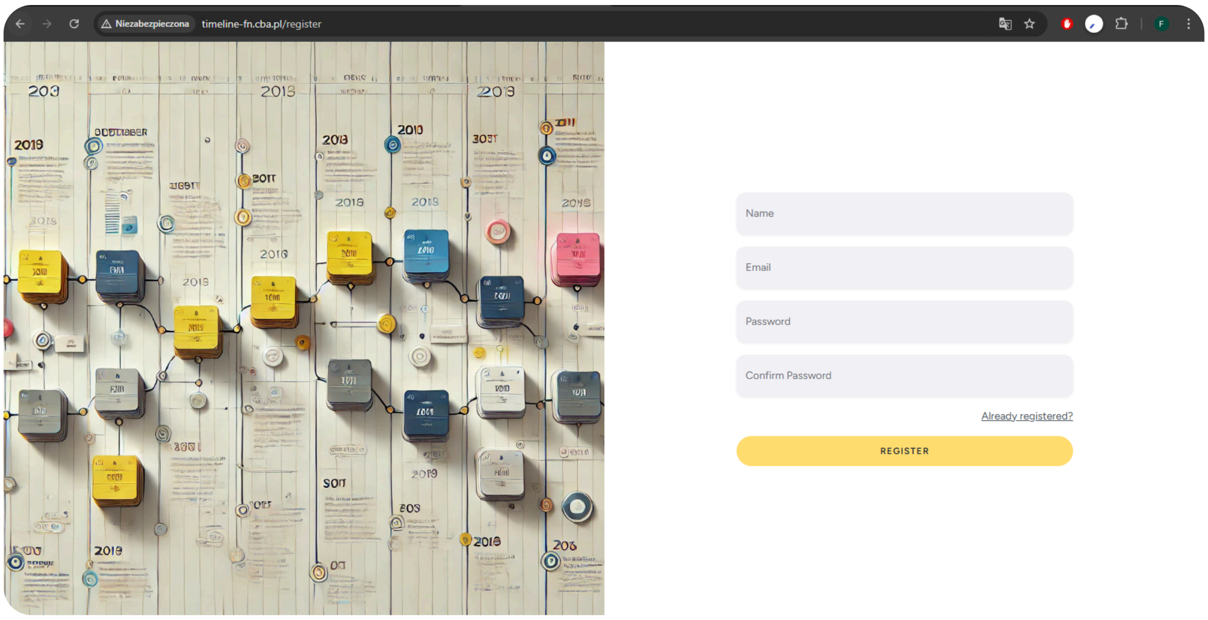
Store, organize, and review your notes and folders in one convenient place.

About Our App

Our app helps you create and organize notes in folders, so you can keep track of ideas, tasks, and goals effortlessly.

Features





Niezabezpieczona timeline-fn.cba.pl/register

Name

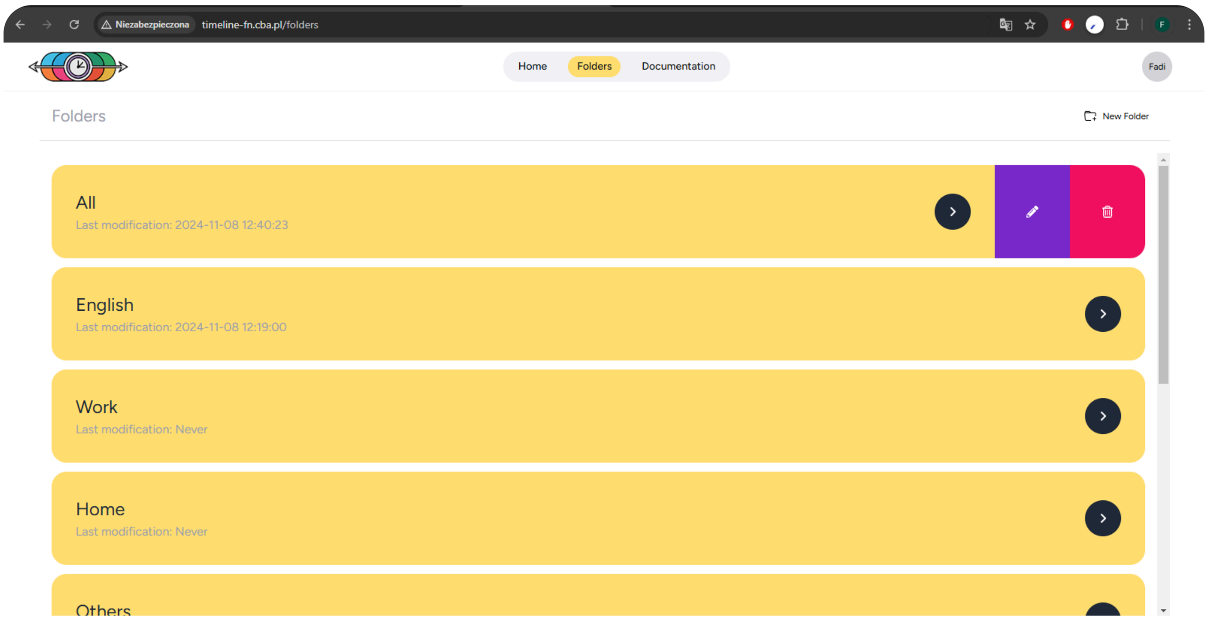
Email

Password

Confirm Password

Already registered?

REGISTER



Niezabezpieczona timeline-fn.cba.pl/folders

Home Folders Documentation Fadi

Folders New Folder

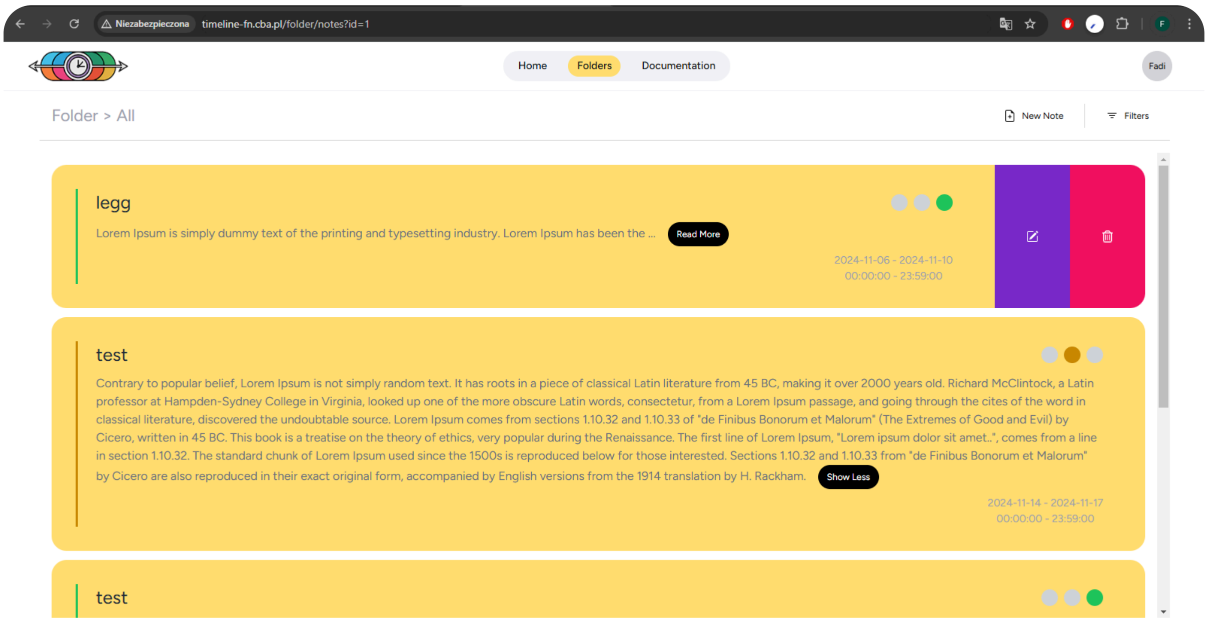
All
Last modification: 2024-11-08 12:40:23

English
Last modification: 2024-11-08 12:19:00

Work
Last modification: Never

Home
Last modification: Never

Others



Niezabezpieczona timeline-fn.cba.pl/folder/notes?id=1

Home Folders Documentation Fadi

Folder > All New Note Filters

legg

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the ... Read More

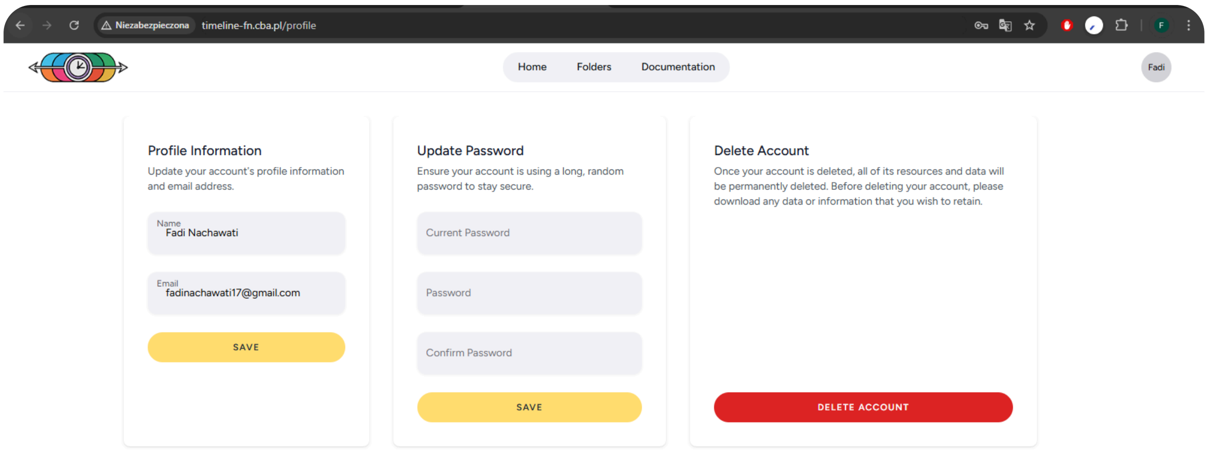
2024-11-06 - 2024-11-10
00:00:00 - 23:59:00

test

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32. The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham. Show Less

2024-11-14 - 2024-11-17
00:00:00 - 23:59:00

test



Profile Information

Update your account's profile information and email address.

Name
Fadi Nachawati

Email
fadinachawati17@gmail.com

SAVE

Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

Password

Confirm Password

SAVE

Delete Account

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

DELETE ACCOUNT