

JOURNALSHAMIRI DOCUMENTATION.

APP

JOURNAL APP DOCUMENTATION.

1. Introduction
2. Features
3. Architecture Overview
4. Setup and Installation
5. Backend - Django
6. Frontend -React Native with Typescript
7. Secure Storage and State Management
8. Authentication
9. API Endpoints
10. Conclusion

Introduction.

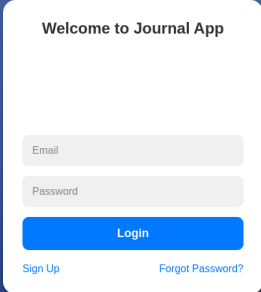
JournalShamiri is a cross-platform mobile application designed for users to manage their daily journal entries. The application includes features for user authentication, journal management, categorization, summary views, and user settings.

Features

1. User Authentication:

- Users can sign up and log in securely using JWT Authentication.
- Passwords are hashed and stored securely.

Login



>Welcome to Journal App

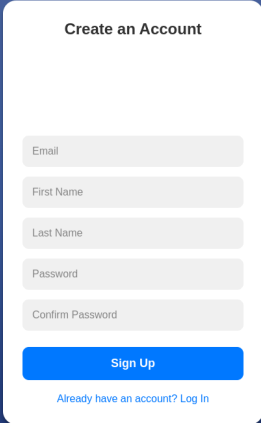
Email

Password

Login

[Sign Up](#) [Forgot Password?](#)

← SignUp



Create an Account

Email

First Name

Last Name

Password

Confirm Password

Sign Up

[Already have an account? Log In](#)

2. Journal Entry Management:

- Add new journal entries with a title, content, category. The date is autosaved in the backend.
- Edit and delete existing entries.

Home

Journal App

Journal Summary (daily)

DailyWeeklyMonthly

Jul 8, 2024: 3 entries

All Categories

This is the Home

Home

Please note that this Ho

We should All Ad

Religion

Let's abide by Good and we will be blessed...

Reporting Date.

Religion

All interns should report to work By monday...

Title

Content

Select Category

Select Category

Work

Home

Religion

Cancel

Update

Delete

Update

Delete

Update

Delete

Home

Journal App

Journal Summary (daily)

DailyWeeklyMonthly

Jul 8, 2024: 3 entries

All Categories

This is the Home

Home

Please note that this Ho

We should All Ad

Religion

Let's abide by Good and we will be blessed...

Reporting Date.

Religion

All interns should report to work By monday...

This is the Home Work

Please note that this Home Work is to be done by the end of the Day. All Houses should be clean

Home

Update Journal

Cancel

Update

Delete

Update

Delete

Update

Delete



Journal Summary (daily)

Daily

Weekly

Monthly

Jul 8, 2024: 3 entries

Add Journal

All Categories

This is the Home Work

Home

Please note that this Home Work is to be done by the end of the Day. All Houses should be clean...

Update

Delete

We should All Adhere to our religious Beleives

Religion

Let's abide by Good and we will be blessed...

Update

Delete

Reporting Date.

Religion

All interns should report to work By monday...

Update

Delete

3. Journal View:

- Users can view a view a list of all their Journal Entries



Journal Summary (daily)

Daily

Weekly

Monthly

Jul 8, 2024: 3 entries

Add Journal

All Categories

This is the Home Work

Home

Please note that this Home Work is to be done by the end of the Day. All Houses should be clean...

Update

Delete

We should All Adhere to our religious Beleives

Religion

Let's abide by Good and we will be blessed...

Update

Delete

Reporting Date.

Religion

All interns should report to work By monday...

Update

Delete

4. Categorization and Summary View:

- Users can categorize their entries, for example personal, work, religion etc.
- Also the app can display a summary of Journal entries over a selected period that is daily, weekly, and montly.



Journal Summary (daily)

Daily

Weekly

Monthly

Jul 8, 2024: 3 entries

Add Journal

All Categories

This is the Home Work

Home

Please note that this Home Work is to be done by the end of the Day. All Houses should be clean...

Update

Delete

We should All Adhere to our religious Beliefs

Religion

Let's abide by Good and we will be blessed...

Update

Delete

Reporting Date.

Religion

All interns should report to work By monday...

Update

Delete

6. Settings:

- Users can update their username and password.

[← ForgotPassword](#)

Reset Password

Email

New Password

Reset Password

[Back to Login](#)

ARCHITECTURE OVERVIEW.

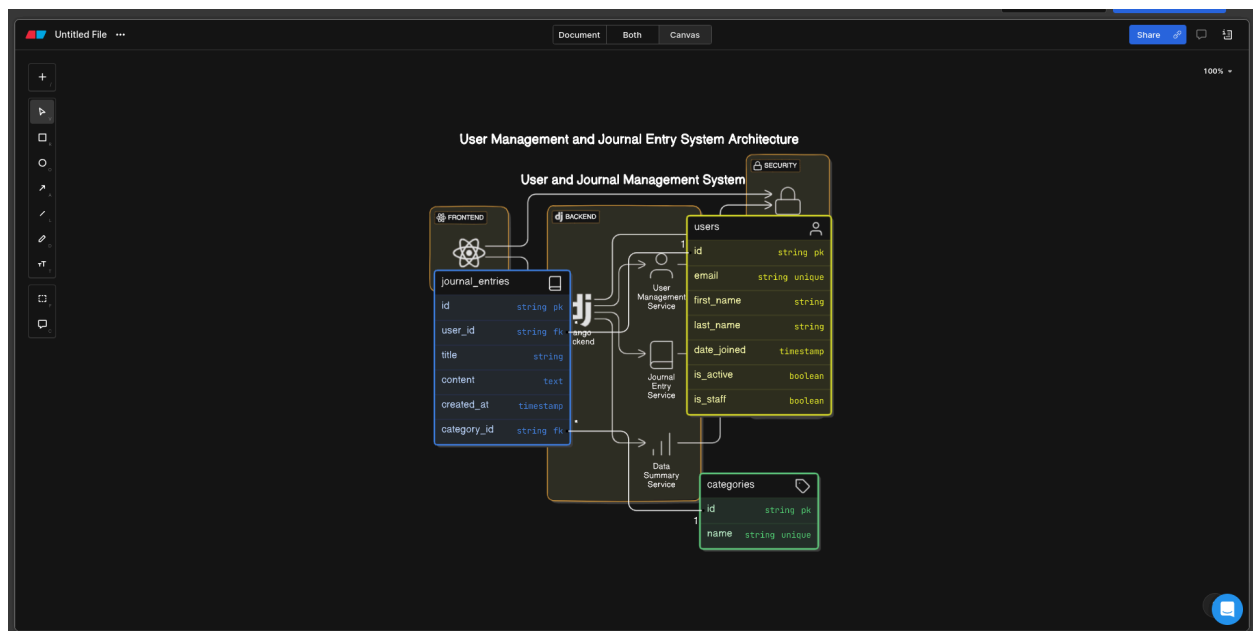
The project follows a client-server architecture:

Backend: Django REST framework with JWT authentication, PostgreSQL database.

Frontend: React Native with TypeScript for mobile development.

State Management: SecureStore and React Redux for secure storage and state management.

The following is the System Architecture



SETUP AND INSTALLATION

Prerequisites:

- Django==4.1.13
- Django REST Framework

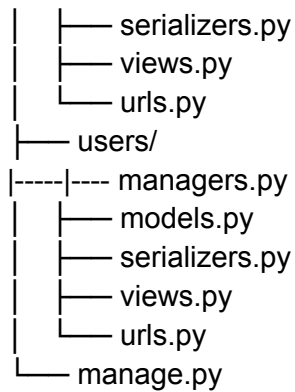
Key dependencies:

- `django rest framework-jwt` for JWT authentication
- `psycopg2` for PostgreSQL database connection

Project Structure.

backend/

```
|— core/
|   |— settings.py
|   |— urls.py
|   |— wsgi.py
|— journal/
|   |— models.py
```



Backend Setup:

1. Clone the repository:

```
git clone https://github.com/Roy-Murithi/JournalShamiri.git
cd JournalShamiri
```

2. Create a virtual environment and install dependencies:

```
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

3. Set up the PostgreSQL database:

```
CREATE DATABASE journal;
```

4. Apply migrations:

```
python manage.py migrate
```

5. python manage.py createsuperuser

```
python manage.py createsuperuser
```

6. Run the development server:

```
python manage.py runserver
```

Frontend Implementation

React Native Setup

- React Native version: X.X.X
- TypeScript version: X.X.X

Key Dependencies

- `@react-navigation/native` for navigation
- `@reduxjs/toolkit` for state management
- `expo-secure-store` for secure storage

Project Structure

```
frontend/  
├── src/  
│   ├── components/  
│   ├── screens/  
│   ├── navigation/  
│   └── context/  
├── App.tsx  
└── package.json
```

Frontend Setup

1. Navigate to the frontend directory:

- `cd journalFrontend`
- `cd ../frontend`

2. Install dependencies:

- `npm install`

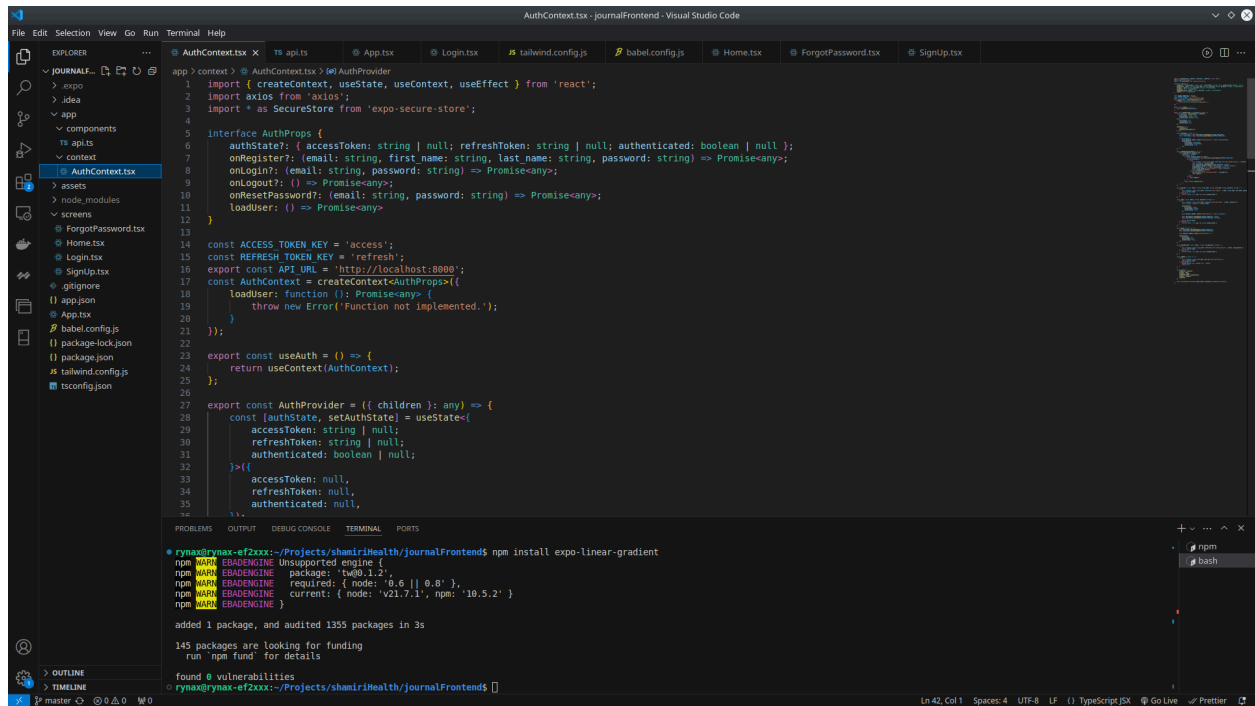
3. Run the React Native application:

- `Npx expo start`

SECURE STORAGE AND STATE MANAGEMENT:

SecureStore: Used to securely store JWT tokens.

React Redux: Used for managing global state.



AUTHENTICATION

- JWT (JSON Web Token) authentication is implemented using `django rest framework-jwt`.
- Tokens are stored securely on the client-side using Expo's SecureStore.

API ENDPOINTS

- `/api/auth/signup/` (POST): User registration
- `/api/auth/login/` (POST): User login
- `/api/journal/entries/` (GET, POST): List and create journal entries
- `/api/journal/entries/<id>/` (GET, PUT, DELETE): Retrieve, update, and delete specific entries
- `/api/journal/categories/` (GET, POST): List and create categories
- `/api/journal/summary/` (GET): Get summary of entries
- `/api/users/profile/` (GET, PUT): Retrieve and update user profile

CONCLUSION

JournalShamiri provides a seamless experience for managing journal entries with secure authentication and efficient data management. The use of Django and React Native ensures a robust and scalable application.