

Todo Mobile App - Full Project Documentation

1. Project Overview

This project is a full-stack Todo Mobile Application built using React Native (Expo) for the frontend and FastAPI (Python) for the backend. It supports Authentication using JWT, CRUD operations on Todos, Profile display, and persistent login using AsyncStorage.

2. Tech Stack

Frontend: React Native, Expo, Axios, React Navigation

Backend: FastAPI, SQLAlchemy, JWT (python-jose), Passlib

Database: SQLite (default) / PostgreSQL (scalable)

3. User Features

- User Signup & Login
- Secure JWT Authentication
- Add, View, Update, Delete Todos
- Mark Todos as Done/Undo
- Profile Page
- Persistent Login

4. Application Flow

1. User signs up or logs in.
2. Token is stored securely in AsyncStorage.
3. Dashboard requests Todos using JWT token.
4. User adds, completes, or deletes todos.
5. Profile page loads user information.

5. Backend API Endpoints

POST /auth/signup -> Create user

POST /auth/login -> Login & get token

GET /me -> Get user profile

GET /todos -> Get user's todos

POST /todos -> Add new todo

PATCH /todos/{id} -> Update todo

DELETE /todos/{id} -> Delete todo

6. Database Design

User Table: id, email, hashed_password

Todo Table: id, title, completed, owner_id

7. Security Implementation

JWT tokens are generated on login.
Passwords are hashed using pbkdf2_sha256.
Protected routes require Authorization header.

8. Error Handling

401: Unauthorized access
422: Validation errors
500: Server errors

9. Project Structure

backend/app/ -> FastAPI files
frontend/src/ -> React Native files
.env -> secret keys (not committed)

10. Testing Checklist

- Signup new user
- Login existing user
- Add todo
- Mark todo done
- Delete todo
- Logout and verify access blocked

11. Submission Ready Features

This project demonstrates Authentication, API Integration, Navigation, CRUD Operations, and Secure Token Storage.