# TCC\_V2.0 Project Architecture Documentation

## Overview

The TCC\_V2.0 project is structured to accommodate various environments (development, production, testing). This documentation explains the folder structure, its purpose, and best practices to maintain consistency.

## Project Structure

Below is the top-level directory structure and their purposes:

|  |  |
| --- | --- |
| Directory | Description |
| app/ | Core application logic (Flask backend) |
| backend/ | Alternative backend directory for different environments (if needed) |
| frontend/ | Frontend UI components, styles, and services |
| database/ | Stores database models and migrations |
| data/ | Static data files (CSV, JSON, backups) |
| migrations/ | Flask-Migrate folder for database schema changes |
| scripts/ | Utility scripts (automation, backups, etc.) |
| uploads/ | Stores uploaded files and user-generated content |
| docs/ | Project documentation and guidelines |
| .env | Stores environment variables (⚠️ Do not commit to Git!) |
| config.py | Configuration file for different environments |
| README.md | Main project documentation |

## Backend Explanation (app/ vs. backend/ Directory)

Which One to Use?

- `app/` is the primary backend directory for local development and testing.

- `backend/` might be used for production deployments or an alternative backend setup.

- If only one backend is needed, remove the unused directory to avoid confusion.

## Frontend Explanation (frontend/ Directory)

Ensure `index.html` or `main.js` exists to load the frontend correctly.

## Database & Migrations (database/ and migrations/)

- `database/` contains models and schema files.

- `migrations/` is used by Flask-Migrate to manage database schema changes.

- Do not store SQLite (`.db`) files inside `app/`. Move them to `database/`.

## Configuration & Environment Handling (config.py & .env)

⚠️ NEVER commit `.env` to Git! Add it to `.gitignore`.

## Deployment Best Practices

✅ Use a `.gitignore` file to exclude unnecessary files.

✅ Ensure the correct backend (`app/` or `backend/`) is used in production.

✅ Move database files (`.db`) to `database/` to prevent accidental deletion.

✅ Set up a clear frontend entry point (`index.html` or `main.js`).

## Final Notes

This project structure is designed for scalability and multi-environment support. If any confusion arises, refer to this document or update it as the project evolves.