TicketFlow – Architecture & Code Structure

Project Overview

TicketFlow is a self-hosted, Jira-like ticket management system built using core PHP and a custom MVC architecture. It is designed to be lightweight, framework-independent, and easy to extend.

Technology Stack

Backend: Core PHP
Architecture: MVC
Database: MySQL (PDO)
Frontend: HTML, Bootstrap, minimal JavaScript
Authentication: Session-based
Routing: Custom Router

High-Level Architecture

Browser requests are routed through a single entry point (public/index.php), processed by a Router, handled by Controllers, executed via Models, and rendered through Views.

Folder Structure

app/: Controllers, Models, Views, Core logic
config/: Application configuration
database/: SQL schema
install/: Installer scripts
public/: Entry point and assets
uploads/: User files

Request Lifecycle

1. Browser sends request
2. public/index.php bootstraps app
3. Router resolves route
4. Controller validates request
5. Model performs DB operations
6. View renders response

Database Design

The database schema is defined in database/ticketflow.sql. Tables are normalized and linked using foreign keys. Prepared statements are used throughout for security.

Security Considerations

PDO prepared statements prevent SQL injection. Session-based authentication is used. Application directories are protected from direct access.

Extension Points

New features can be added by extending controllers, models, or views without modifying core files.

Known Limitations

No middleware layer, no REST API, no automated tests yet.

Future Improvements

Service layer, RBAC, REST APIs, audit logging, automated testing, enhanced documentation.