ASCapstone2025

Technical Design & Code Reference Documentation

Version: 1.0

Date: 2025-09-02

Author: BRA MEDE

1. Overview

ASCapstone2025 is a PHP-based web application designed to manage customer profiles, administrative tasks, user accounts, and provide a public-facing homepage. It features a modular architecture with views, models, assets, and controllers for maintainability and scalability.

2. Architecture

The application follows a loose MVC-like pattern with PHP views acting as both controllers and renderers. Frontend: - HTML, CSS, JavaScript (Bootstrap, jQuery, TinyMCE, Select2) - Views: homeView.php, adminView.php, loginView.php, etc. - Sections: header, carousel, reviews, services, map, footer Backend: - Models in /models/: db.php, model_admin.php, customer_profiles.php - Database configuration: dbconfig.ini - File uploads handled by upload.php - Session-based authentication Assets: - Images (static in /images, uploads in /bcimage) - CSS/JS for styling and interactivity

3. Database Design

Database Design: - customers(ID, firstName, lastName, phoneNumber, email, street, apt, city, state, zipcode, serviceRequested, notes) - adminlogin(username, password (hashed), adminEmail) - bcimage(id, filename, path) [inferred] Configuration stored in models/dbconfig.ini. PDO used for secure queries with prepared statements.

4. Code Reference

models/db.php: - $ini: array of DB configuration - $db: PDO connection instance models/model_admin.php functions: - login(username, password): bool - getAllCustomers(): array - getCustomer(ID): array - deleteCustomer(customerID): string - addCustomer(customerData): bool - updateCustomer(customerData): bool - editCustomer(customerID): array - searchCustomer(searchTerm): array - getAllAdmins(): array - register(username, password, email): string - updateAccount(username, email, currentPassword, newPassword): string - getAdmin(username): array|null - requestQuote(firstName, lastName, email, phoneNumber, street, apt, city, state, zip, serviceRequested, notes): string models/customer_profiles.php: - Currently minimal/empty, reserved for future. JavaScript (adminScript.js): - showAddCustomerForm(): Prompt & submit form for customer creation - showEditCustomerForm(customer): Prefill form and submit - initEditableSection({editButtonId, textElementId, formId, editorId}): Enable TinyMCE editor on section HTML Forms: - Quote form fields: firstName, lastName, email, phone, service-requested, notes - Login form: username, password - Admin update: username, email, currentPassword, newPassword

5. Security

Security considerations: - Sessions for authentication (regenerate on login) - Passwords currently SHA1+salt (upgrade to password_hash) - Input validation in PHP & JS - PDO prepared statements - File uploads validated for type/size (recommend stricter MIME checks) - Recommended: HTTPS in production, CSRF tokens, secure headers (CSP, HSTS, etc.)

6. Future Improvements

- Add RESTful API endpoints - Implement roles & permissions for admins - Enhance frontend with React/Vue - Stronger password hashing (bcrypt/Argon2) - Add automated PHPUnit tests - Optimize DB queries and enable caching - Monitoring & audit logging