Procurement Admin Portal - Technical Documentation

Table of Contents

1. Overview

The Procurement Admin Portal is a comprehensive web application designed to manage the entire procurement process for Ahmedabad University. It facilitates vendor management, project creation, bid submission, evaluation, and award processes with a focus on efficiency, transparency, and security.

Vendor Management

Approve and manage vendor profiles, review documents, and track vendor activities.

Procurement Projects

Create, manage, and track procurement projects with detailed specifications and requirements.

Bid Management

Evaluate vendor bids, manage technical and financial reviews, and award projects.

Content Management

Customize email templates, terms & conditions, and other content for the portal.

> **Note:** This portal is designed to work alongside a client-facing vendor portal where vendors can register, submit bids, and track their participation in procurement processes.

2. System Architecture

The Procurement Admin Portal follows a Model-View-Controller (MVC) architecture pattern with additional layers for security, middleware, and helpers. The application is built using PHP, with a focus on modular, secure, and maintainable code.

Architecture Diagram

```
┌─────────────────────────────────┐
│          Client Browser          │
└───────────────┬─────────────────┘
                │
┌───────────────▼─────────────────┐
│        index.php (Entry)         │
└───────────────┬─────────────────┘
                │
┌───────────────▼─────────────────┐
│             Router               │
└───────────────┬─────────────────┘
                │
┌───────────────▼─────────────────┐
│          Middlewares             │
│  - Authentication                │
│  - JWT Auth                      │
│  - Data Sanitization             │
│  - Permission                    │
└───────────────┬─────────────────┘
                │
┌───────────────▼─────────────────┐
│          Controllers             │
└───────────────┬─────────────────┘
                │
┌──────┬────────▼────────┬─────────┐
│      │                 │         │
▼      ▼                 ▼         ▼
Models  Views          Helpers    Core
```

Data Flow

1. **Request Initiation:** All requests are processed through the `index.php` file.
2. **Routing:** The `Router` class determines which controller and action to call based on the URL.
3. **Middleware Processing:** Request passes through middleware layers for authentication, data sanitization, and permission checks.
4. **Controller Processing:** The appropriate controller processes the request and interacts with the models.

5. **Data Retrieval/Persistence:** Models interact with the database via the `QueryBuilder` and `Database` classes.
6. **View Rendering:** Controllers pass data to views for rendering the HTML response.
7. **Response:** The final HTML is sent back to the client.

3. Technology Stack

Backend

- **PHP:** Server-side scripting language
- **MySQL:** Relational database management system
- **PDO:** PHP Data Objects for database access

Frontend

- **HTML5/CSS3:** Markup and styling
- **JavaScript:** Client-side scripting
- **jQuery:** JavaScript library for DOM manipulation
- **Bootstrap:** CSS framework for responsive design
- **Tailwind CSS:** Utility-first CSS framework
- **Remix Icon:** Icon library
- **DataTables:** jQuery plugin for table functionality
- **Summernote:** WYSIWYG editor

Security

- **JWT (JSON Web Tokens):** For secure authentication
- **OAuth 2.0:** For Google authentication
- **Session Encryption:** For secure session storage
- **Content Security Policy (CSP):** For mitigating XSS attacks
- **CSRF Protection:** For cross-site request forgery prevention

Libraries & Tools

- **TCPDF:** For PDF generation
- **Mailjet API:** For email sending
- **Google OAuth API:** For authentication

4. Project Structure

The project follows a modular structure with clear separation of concerns:

```
procadmin/
├── app/                    # Application code
│   ├── config/             # Configuration files
│   ├── controllers/        # MVC Controllers
│   ├── core/               # Core functionality
│   ├── helpers/            # Helper classes
│   ├── libs/               # Third-party libraries
│   ├── middleware/         # Middleware components
│   ├── models/             # MVC Models
│   └── views/              # MVC Views
├── public/                 # Publicly accessible files
│   └── assets/             # CSS, JS, images, fonts
├── index.php               # Application entry point
└── README.md               # Project documentation
```

Key Directories & Files

App Configuration

- `app/config/config.php` : Main configuration file with application settings
- `app/config/oauth_config.php` : OAuth configuration for Google authentication
- `app/config/Database.php` : Database connection configuration

Controllers

- `AdminBaseController.php` : Base controller with common functionality
- `AdminAuthController.php` : Authentication management
- `AdminDashboardController.php` : Dashboard and analytics
- `AdminProcurementController.php` : Procurement project management
- `AdminVendorController.php` : Vendor management
- `AdminUserManagementController.php` : User and role management
- `AdminContentController.php` : Content management
- `AdminDepartmentCategoryController.php` : Departments and categories
- `ErrorController.php` : Error handling

Core Components

- `App.php` : Main application initialization
- `Router.php` : URL routing system
- `Security.php` : Security utilities
- `SessionEncryption.php` : Encrypted session management
- `QueryBuilder.php` : SQL query builder

- `PermissionHandler.php` : Permission management
- `MiddlewareHandler.php` : Middleware pipeline
- `CacheManager.php` : Caching system
- `ViewEngine.php` : View rendering engine

Models

- `BaseModel.php` : Base model with common functionality
- `AdminUserModel.php` : Admin user management
- `AdminRoleModel.php` : Role management
- `AdminPermissionModel.php` : Permission management
- `ProjectModel.php` : Procurement project management
- `BidModel.php` : Bid management
- `UserProfileModel.php` : Vendor profile management
- `DepartmentModel.php` : Department management
- `BusinessCategoryModel.php` : Business category management
- `AdminTokenModel.php` : JWT token management
- `DashboardModel.php` : Dashboard analytics

Middleware

- `Middleware.php` : Base middleware class
- `AdminAuthMiddleware.php` : Authentication check
- `JWTAuthMiddleware.php` : JWT token verification
- `DataSanitizationMiddleware.php` : Input sanitization
- `PermissionMiddleware.php` : Permission check

Helpers

- `AuthHelper.php` : Authentication utilities
- `JWTHelper.php` : JWT token utilities
- `GoogleOAuthHelper.php` : Google OAuth utilities
- `DataEncryptionHelper.php` : Data encryption utilities
- `SecureUrlHelper.php` : Secure URL generation
- `ExportHelper.php` : Data export utilities
- `FileHelper.php` : File handling utilities
- `MailService.php` : Email sending service
- `StringHelper.php` : String manipulation utilities

Views

- `layouts/` : Reusable layout templates
- `auth/` : Authentication views
- `dashboard/` : Dashboard views
- `procurement/` : Procurement management views
- `vendor/` : Vendor management views
- `usermanagement/` : User management views

- `content/`: Content management views
- `settings/`: Settings views
- `errors/`: Error views

5. Database Schema

Entity-Relationship Diagram

```
┌─────────────────┐      ┌───────────────────┐      ┌─────────────────┐
│  admin_users    │      │   admin_roles     │      │ admin_permissions│
└─────────────────┘      └───────────────────┘      └─────────────────┘
        │                          │                          │
        │                          │                          │
        └──────────────────────────┼──────────────────────────┘
                                   │
┌─────────────────┐      ┌───────────────────┐      ┌─────────────────┐
│     users       │──────│       bids        │──────│    projects     │
└─────────────────┘      └───────────────────┘      └─────────────────┘
        │                                                     │
        │                                                     │
┌─────────────────┐                                 ┌─────────────────┐
│  user_profiles  │                                 │   departments   │
└─────────────────┘                                 └─────────────────┘
        │
        │
┌─────────────────┐      ┌───────────────────┐
│user_documents   │      │  user_references  │
└─────────────────┘      └───────────────────┘
```

Main Database Tables

Admin Management

- **admin_users:** Admin user accounts
- **admin_roles:** Admin user roles
- **admin_permissions:** Permission definitions
- **admin_role_permissions:** Role-permission mappings
- **admin_tokens:** JWT tokens for authentication

Vendor Management

- **users:** User accounts (vendors)
- **user_profiles:** Detailed vendor profiles
- **user_documents:** Vendor uploaded documents
- **user_references:** Vendor references
- **business_categories:** Business category definitions

Procurement Management

- **projects:** Procurement projects
- **project_requirements:** Project requirements

- **project_specifications:** Technical specifications
- **project_documents:** Project documents
- **project_milestones:** Project milestones
- **project_questionnaire:** Technical questionnaire
- **project_tags:** Project tags for categorization
- **project_tag_mapping:** Project-tag mappings
- **departments:** Department definitions

Bid Management

- **bids:** Vendor bids on projects
- **bid_questionnaire_answers:** Answers to technical questionnaires

Authentication & Security

- **auth_tokens:** Authentication tokens for vendors
- **otp_verifications:** OTP for email verification

Key Relationships

- Admin users have roles, which have permissions
- Projects belong to departments
- Bids link users (vendors) to projects
- User profiles are associated with users (vendors)
- Projects have various related entities (requirements, specifications, documents, etc.)

6. Core Components

App.php

Central application class responsible for initializing the application, processing requests, and managing the application lifecycle.

- Initializes secure sessions
- Sets up security components
- Manages routing
- Handles middleware pipeline
- Executes controller actions

Router.php

Routes incoming requests to the appropriate controller and action.

- Defines route mappings
- Handles dynamic routes with parameters
- Associates routes with permissions
- Supports secure URL generation

Security.php

Provides security utilities for the application.

- XSS protection
- CSRF token generation and validation
- Content Security Policy (CSP) implementation
- Data encryption/decryption
- Input sanitization

SessionEncryption.php

Manages encrypted sessions for enhanced security.

- Session data encryption
- Session cookie security settings
- Session garbage collection

QueryBuilder.php

Provides a fluent interface for building SQL queries.

- SQL query building with method chaining
- SQL injection prevention with prepared statements
- Advanced query conditions
- Transaction support
- Query caching

BaseModel.php

Base model class with common database functionality.

- CRUD operations
- Query execution
- Data retrieval with conditions
- Permission integration

PermissionHandler.php

Manages role-based access control.

- Permission checking
- Permission-based action execution
- Access denied handling

AdminBaseController.php

Base controller with common functionality for all admin controllers.

- View rendering
- Redirection
- JSON response handling
- Authentication checking
- JWT token refreshing
- Permission checking

ViewEngine.php

View rendering engine for the MVC architecture.

- View loading and rendering
- Data passing to views
- Layout management
- Helper methods for views

7. Key Features

Authentication & Authorization

- **Google OAuth Integration:** Secure authentication using Google accounts
- **JWT Authentication:** Token-based authentication with automatic refreshing
- **Role-Based Access Control:** Fine-grained permissions for admin users
- **Session Encryption:** Encrypted session storage for sensitive data

Vendor Management

- **Vendor Approval:** Review and approve vendor registrations
- **Document Verification:** Review and verify vendor documents
- **Status Management:** Manage vendor status (active, pending, inactive)
- **Vendor Export:** Export vendor data to CSV and PDF formats

Procurement Management

- **Project Creation:** Create detailed procurement projects
- **Technical Specifications:** Define technical requirements
- **Document Management:** Upload and manage project documents
- **Milestone Tracking:** Track project milestones
- **Project Export:** Export project data to CSV and PDF formats

Bid Management

- **Bid Review:** Review and evaluate vendor bids
- **Technical Evaluation:** Technical evaluation of bids
- **Financial Evaluation:** Financial evaluation of bids
- **Award Process:** Award bids to selected vendors
- **Bid Export:** Export bid data to CSV and PDF formats

User & Role Management

- **User Management:** Create and manage admin users
- **Role Management:** Create and manage roles with specific permissions
- **Permission Management:** Fine-grained permission control
- **Access Control:** Restrict access to specific functionality based on roles

Content Management

- **Email Templates:** Customize email templates for various notifications
- **Terms & Conditions:** Manage terms and conditions content
- **Home Page Content:** Customize home page content for the client portal

Dashboard & Analytics

- **Project Statistics:** View project statistics (total, active, completed)
- **Bid Statistics:** View bid statistics (total, pending, awarded)
- **Vendor Statistics:** View vendor statistics (verified, pending)
- **Recent Activities:** View recent projects, bids, and vendors
- **Upcoming Deadlines:** View projects with upcoming deadlines

File Handling & Export

- **Document Upload:** Secure file upload for project documents
- **CSV Export:** Export data to CSV format
- **PDF Export:** Generate PDF reports for projects, bids, and vendors

Email Notifications

- **Vendor Approval:** Notify vendors when their profiles are approved
- **Bid Status Updates:** Notify vendors about bid status changes
- **OTP Verification:** Send OTP for email verification
- **Bid Confirmation:** Confirm bid submissions to vendors

8. Authentication & Authorization

Authentication Flow

1. **Google OAuth:** Admin users authenticate using Google OAuth 2.0
2. **Email Domain Validation:** Only users with allowed email domains can access the admin portal
3. **JWT Token Generation:** Upon successful authentication, JWT access and refresh tokens are generated
4. **Session Storage:** Tokens are stored in encrypted sessions
5. **Token Verification:** Each request is verified using the JWT token
6. **Automatic Token Refresh:** Expired tokens are automatically refreshed using the refresh token

Authorization System

The application uses a role-based access control (RBAC) system with the following components:

Roles
Roles are defined in the `admin_roles` table and represent different admin user types:

- **Super Admin:** Full access to all functionality
- **Procurement Manager:** Manage procurement projects and bids
- **Vendor Manager:** Manage vendors and their profiles
- **Content Manager:** Manage content and email templates
- **Viewer:** View-only access to data

Permissions
Permissions are defined in the `admin_permissions` table and represent specific actions:

- **users.view:** View user management
- **users.create:** Create admin users
- **users.edit:** Edit admin users
- **users.delete:** Delete admin users
- **roles.manage:** Manage roles and permissions
- **vendors.view:** View vendor management
- **vendors.status:** Change vendor status
- **vendors.delete:** Delete vendors
- **vendors.export:** Export vendor data
- **procurement.view:** View procurement projects
- **procurement.create:** Create procurement projects
- **procurement.edit:** Edit procurement projects

- **procurement.delete:** Delete procurement projects
- **procurement.export:** Export procurement data
- **bids.view:** View bids
- **bids.status:** Change bid status
- **bids.award:** Award bids
- **bids.reject:** Reject bids
- **bids.export:** Export bid data
- **content.manage:** Manage content
- **departments.manage:** Manage departments
- **categories.manage:** Manage business categories

Permission Checking
Permission checks are performed at multiple levels:

1. **Router Level:** Routes are associated with required permissions
2. **Middleware Level:** `PermissionMiddleware` checks permissions before executing controller actions
3. **Controller Level:** `requirePermission()` method checks permissions within controllers
4. **Model Level:** `withPermission()` method checks permissions before executing database operations
5. **View Level:** `AuthHelper::hasPermission()` checks permissions in views to conditionally render elements

9. Security Features

Authentication Security

- **OAuth 2.0:** Secure authentication using Google accounts
- **JWT Tokens:** Signed JWT tokens for secure authentication
- **Token Expiration:** Access tokens expire after 1 hour, refresh tokens after 30 days
- **Token Revocation:** Tokens can be revoked for security reasons
- **Session Encryption:** Session data is encrypted using AES-256-CBC

Input Validation & Sanitization

- **Input Sanitization:** All input is sanitized using `DataSanitizationMiddleware`
- **XSS Protection:** HTML special characters are escaped to prevent XSS attacks
- **SQL Injection Prevention:** Prepared statements are used for all database queries
- **Data Type Validation:** Input data types are validated to prevent type-related attacks

CSRF Protection

- **CSRF Tokens:** All forms include CSRF tokens
- **Token Validation:** CSRF tokens are validated for all POST requests
- **Token Generation:** Secure random token generation using `random_bytes()`

Content Security Policy (CSP)

- **Script Source Restriction:** Scripts are only allowed from trusted sources
- **Style Source Restriction:** Styles are only allowed from trusted sources
- **Frame Ancestors:** Frame ancestors are restricted to prevent clickjacking
- **Object Sources:** Object sources are restricted to prevent object injection
- **Form Actions:** Form actions are restricted to prevent form-based attacks

Secure Headers

- **X-Content-Type-Options:** Prevents MIME type sniffing
- **X-Frame-Options:** Prevents clickjacking
- **X-XSS-Protection:** Enables browser XSS protection
- **Referrer-Policy:** Controls referrer information
- **Permissions-Policy:** Restricts browser permissions

Data Encryption

- **Sensitive Data Encryption:** Sensitive vendor data is encrypted in the database
- **JWT Token Security:** JWT tokens are signed with a secure key
- **Session Encryption:** Session data is encrypted
- **Secure Cookie Handling:** Cookies have secure and httpOnly flags

Error Handling & Logging

- **Secure Error Handling:** Detailed errors are logged but not shown to users
- **Custom Error Pages:** Custom error pages for various error scenarios
- **Error Logging:** Errors are logged for debugging and security analysis

10. Frontend Implementation

Layout Structure

The frontend uses a responsive layout with the following components:

- **Header:** Top navigation with user menu
- **Sidebar:** Left sidebar with main navigation
- **Main Content:** Main content area for each page
- **Footer:** Page footer with copyright information

Responsive Design

The application is responsive and works on various screen sizes:

- **Desktop:** Full layout with sidebar
- **Tablet:** Collapsible sidebar
- **Mobile:** Mobile-friendly layout with collapsible sidebar and simplified navigation

User Interface Components

- **Forms:** Standardized forms for data entry
- **Tables:** DataTables for data display with sorting, filtering, and pagination
- **Modals:** Modal dialogs for interactive actions
- **Cards:** Card layout for dashboard widgets
- **Alerts:** Alert messages for user notifications
- **Buttons:** Styled buttons for actions
- **Icons:** Remix Icon library for icons

Color Scheme

The application uses a consistent color scheme defined in CSS variables:

```
:root {
    --background-primary: #FFFFFF;
    --background-secondary: #F0F0F0;
    --background-dark: #424242;
    --text-primary: #212529;
    --text-secondary: #565161;
    --text-light: #FFFFFF;
    --text-accent: #801214;
    --accent-red: #801214;
    --accent-dark: #333333;
    --accent-gray: #494E55;
    --accent-light: #D6D6D6;
}
```

JavaScript Functionality

- **AJAX Requests:** Asynchronous data loading and submission
- **Form Validation:** Client-side form validation
- **DataTables:** Enhanced table functionality
- **Summernote Editor:** WYSIWYG editing for content management
- **JWT Token Refresh:** Automatic JWT token refresh for seamless authentication

Custom Components

- **Project Form:** Complex form for project creation and editing
- **Bid Details:** Detailed bid view with technical and financial information
- **Vendor Profile:** Comprehensive vendor profile with documents and references
- **Email Template Editor:** Custom editor for email templates
- **Terms & Conditions Editor:** Structured editor for terms and conditions

11. API Endpoints

Authentication Endpoints

Route	Controller	Action	Description
/login	AdminAuthController	login	Login page
/google-auth	AdminAuthController	googleAuth	Redirect to Google OAuth
/google-callback	AdminAuthController	handleGoogleCallback	Google OAuth callback
/refresh-token	AdminAuthController	refreshToken	Refresh JWT token
/logout	AdminAuthController	logout	Logout and session destruction

User Management Endpoints

Route	Controller	Action	Description
/user-management	AdminUserManagementController	index	User management page
/create-user	AdminUserManagementController	create	Create user (AJAX)
/update-user	AdminUserManagementController	update	Update user (AJAX)
/delete-user	AdminUserManagementController	delete	Delete user (AJAX)
/user-details-{id}	AdminUserManagementController	details	User details page
/user-roles	AdminUserManagementController	roles	Role management page
/create-role	AdminUserManagementController	createRole	Create role (AJAX)
/update-role	AdminUserManagementController	updateRole	Update role (AJAX)
/delete-role	AdminUserManagementController	deleteRole	Delete role (AJAX)
/get-role-permissions	AdminUserManagementController	getRolePermissions	Get role permissions (AJAX)

Vendor Management Endpoints

Route	Controller	Action	Description
/vendors	AdminVendorController	index	Vendor management page
/vendor-details-{id}	AdminVendorController	details	Vendor details page
/vendor-update-status	AdminVendorController	updateStatus	Update vendor status (AJAX)
/vendor-delete	AdminVendorController	delete	Delete vendor (AJAX)
/export-vendors	AdminVendorController	exportVendors	Export vendors to CSV
/export-vendor-pdf-{id}	AdminVendorController	exportVendorPdf	Export vendor to PDF

Procurement Management Endpoints

Route	Controller	Action	Description
/procurement-management	AdminProcurementController	index	Procurement management page
/procurement-create	AdminProcurementController	create	Create project (AJAX)
/procurement-update	AdminProcurementController	update	Update project (AJAX)
/procurement-delete	AdminProcurementController	delete	Delete project (AJAX)
/procurement-details-{id}	AdminProcurementController	details	Project details page
/procurement-update-status	AdminProcurementController	updateStatus	Update project status (AJAX)
/get-project-data	AdminProcurementController	getProjectData	Get project data (AJAX)
/update-milestone-status	AdminProcurementController	updateMilestoneStatus	Update milestone status (AJAX)
/export-projects	AdminProcurementController	exportProjects	Export projects to CSV
/export-project-pdf-{id}	AdminProcurementController	exportProjectPdf	Export project to PDF
/export-project-bids-{id}	AdminProcurementController	exportProjectBids	Export project bids to CSV
/export-all-bids	AdminProcurementController	exportAllBids	Export all bids to CSV

Bid Management Endpoints

Route	Controller	Action	Description
/bid-details-{id}	AdminProcurementController	bidDetails	Bid details page
/bid-update-status	AdminProcurementController	updateBidStatus	Update bid status
/bid-award	AdminProcurementController	awardBid	Award bid
/bid-reject	AdminProcurementController	rejectBid	Reject bid
/export-bid-pdf-{id}	AdminProcurementController	exportBidPdf	Export bid to PDF

Content Management Endpoints

Route	Controller	Action	Description
/content-management	AdminContentController	index	Content management page
/edit-template-{id}	AdminContentController	edit	Edit email template
/save-template	AdminContentController	save	Save email template
/edit-terms	AdminContentController	editTerms	Edit terms & conditions
/save-terms	AdminContentController	saveTerms	Save terms & conditions
/edit-home	AdminContentController	editHome	Edit home page content
/save-home	AdminContentController	saveHome	Save home page content

Department & Category Management Endpoints

Route	Controller	Action	Description
/department-category	AdminDepartmentCategoryController	index	Department & category management page
/save-department	AdminDepartmentCategoryController	saveDepartment	Save department
/save-business-category	AdminDepartmentCategoryController	saveBusinessCategory	Save business category
/delete-department	AdminDepartmentCategoryController	deleteDepartment	Delete department
/delete-business-category	AdminDepartmentCategoryController	deleteBusinessCategory	Delete business category

Error Handling Endpoints

Route	Controller	Action	Description
/access-denied	ErrorController	accessDenied	Access denied page
/not-found	ErrorController	notFound	Not found page
/server-error	ErrorController	serverError	Server error page

12. Deployment Guidelines

Server Requirements

- **PHP:** PHP 7.4 or higher
- **MySQL:** MySQL 5.7 or higher
- **Apache/Nginx:** Web server with URL rewriting support
- **SSL:** SSL certificate for HTTPS
- **PHP Extensions:** PDO, PDO_MySQL, OpenSSL, mbstring, json, fileinfo

Deployment Steps

1. **Database Setup:**
    - Create MySQL database
    - Import database schema
    - Set up initial admin user
2. **Application Setup:**
    - Upload application files to server
    - Set proper permissions for directories
    - Create upload directories
3. **Configuration:**
    - Configure database connection in `app/config/config.php`
    - Configure Google OAuth in `app/config/oauth_config.php`
    - Set up Mailjet API keys
    - Configure application URL and paths
4. **Web Server Configuration:**
    - Set up URL rewriting rules
    - Configure SSL certificate
    - Set up proper document root
5. **Security Configuration:**
    - Generate secure application key
    - Set proper file permissions
    - Configure secure headers
6. **Testing:**
    - Test authentication flow
    - Test core functionality
    - Test file uploads and exports

Directory Permissions

- **app/logs/:** Writable (755)
- **app/cache/:** Writable (755)
- **public/assets/:** Readable (644)
- **procuploads/uploadtest/:** Writable (755) for file uploads

Security Recommendations

- **HTTPS:** Use HTTPS for all communications
- **Secure Headers:** Implement secure HTTP headers
- **Strong Passwords:** Enforce strong passwords for admin users
- **Regular Updates:** Keep PHP, MySQL, and other components up to date
- **Backup Strategy:** Implement regular backups of database and uploads
- **Access Control:** Restrict access to sensitive directories
- **Database Security:** Use separate database user with minimal privileges
- **File Upload Security:** Validate and sanitize all file uploads
- **Error Handling:** Configure proper error handling to prevent information disclosure

Environment-Specific Configuration

The application supports different environment configurations using the `ENVIRONMENT` constant in `config.php`:

```
// Environment setting
define('ENVIRONMENT', 'development'); // Options: 'development', 'testing'
```

Development Environment

- Display detailed error messages
- Disable caching
- Use development database
- Enable debug logging

Production Environment

- Hide detailed error messages
- Enable caching
- Use production database
- Disable debug logging
- Enforce strict security settings

Maintenance

- **Cache Management:** Clear cache periodically or when making significant changes
- **Log Rotation:** Set up log rotation to prevent log files from growing too large
- **Database Maintenance:** Perform regular database maintenance and optimization
- **Token Cleanup:** Run periodic cleanup of expired tokens from the database
- **Security Updates:** Regularly update dependencies and security patches

Troubleshooting

- **Check Logs:** Application logs are stored in `app/logs/`
- **Clear Cache:** Clear cache by accessing `?clear_cache=APP_KEY`
- **Database Connection:** Verify database connection settings in `config.php`
- **OAuth Configuration:** Verify Google OAuth settings in `oauth_config.php`
- **File Permissions:** Check permissions on upload and cache directories
- **Session Issues:** Check session configuration and cookie settings

Ahmedabad University Procurement Portal - Technical Documentation