

PHP Backend Setup Instructions

Directory Structure

Here's where to place the PHP files in your Svelte project:

Plain Text

```
office-of-the-dead/
├── src/
│   ├── routes/
│   │   └── purchase/
│   │       ├── +page.svelte (original)
│   │       └── +page-updated.svelte (with PHP backend)
│   ├── static/
│   │   └── api/
│   │       ├── submit-commission.php (main handler)
│   │       └── config.php (database config)
│   └── database/
│       ├── setup.sql (database schema)
│       └── PHP-SETUP-INSTRUCTIONS.md (this file)
```

Step-by-Step Setup

1. Database Setup

Create the database:

Bash

```
mysql -u root -p < database/setup.sql
```

Or manually in MySQL:

SQL

```
CREATE DATABASE office_of_the_dead CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
USE office_of_the_dead;
-- Then run the contents of database/setup.sql
```

2. Configure Database Connection

Edit static/api/config.php :

PHP

```
return [  
    'database' => [  
        'host' => 'localhost',           // Your MySQL host  
        'dbname' => 'office_of_the_dead', // Database name  
        'username' => 'your_username',   // Your MySQL username  
        'password' => 'your_password',   // Your MySQL password  
        'charset' => 'utf8mb4'  
    ],  
    // ... rest of config  
];
```

3. Update the Svelte Component

Replace the purchase page:

Bash

```
# Backup original  
mv src/routes/purchase/+page.svelte src/routes/purchase/+page-original.svelte  
  
# Use the updated version with PHP backend  
mv src/routes/purchase/+page-updated.svelte src/routes/purchase/+page.svelte
```

4. Server Requirements

Your web server needs:

- **PHP 7.4+** with PDO MySQL extension
- **MySQL 5.7+** or **MariaDB 10.2+**
- **Web server** (Apache/Nginx) configured to serve PHP files

5. File Permissions

Set proper permissions for the PHP files:

Bash

```
chmod 644 static/api/*.php  
chmod 600 static/api/config.php # More restrictive for config
```

6. Testing the Setup

Test database connection:

Create a test file `static/api/test-db.php` :

PHP

```
<?php
$config = require 'config.php';
try {
    $pdo = new PDO(
        "mysql:host={$config['database']['host']};dbname={$config['database']
['dbname']};charset={$config['database']['charset']}",
        $config['database']['username'],
        $config['database']['password']
    );
    echo "Database connection successful!";
} catch (PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

Visit `http://yourdomain.com/api/test-db.php` to test.

Deployment Considerations

For Shared Hosting

1. **Upload files** to your web hosting account
2. **Create database** through your hosting control panel
3. **Import** the `database/setup.sql` file
4. **Update** `static/api/config.php` with your hosting database credentials

For VPS/Dedicated Server

1. **Install LAMP/LEMP stack:**

Bash

```
# Ubuntu/Debian
sudo apt update
sudo apt install apache2 mysql-server php php-mysql php-pdo
```

```
# CentOS/RHEL
```

```
sudo yum install httpd mariadb-server php php-mysqlnd php-pdo
```

1. **Configure virtual host** to serve your Svelte build files
2. **Set up SSL certificate** (Let's Encrypt recommended)

Security Considerations

1. **Move config outside web root:**

PHP

```
// In submit-commission.php, change:  
$config = require '../.../config/database.php';  
// Instead of:  
$config = require 'config.php';
```

1. **Add input validation and sanitization**
2. **Implement rate limiting** to prevent spam
3. **Use prepared statements** (already implemented)
4. **Enable HTTPS** for all form submissions

Database Management

View Submissions

SQL

```
SELECT  
  id,  
  CONCAT(first_name, ' ', last_name) as full_name,  
  email,  
  amount,  
  newsletter,  
  intentions,  
  created_at,  
  processed  
FROM prayer_commissions  
ORDER BY created_at DESC;
```

Mark as Processed

SQL

```
UPDATE prayer_commissions
SET processed = TRUE, processed_at = NOW()
WHERE id = ?;
```

Export Data

SQL

```
SELECT * FROM prayer_commissions
INTO OUTFILE '/tmp/commissions.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Troubleshooting

Common Issues

"Database connection failed"

- Check database credentials in `config.php`
- Verify MySQL service is running
- Ensure database exists and user has proper permissions

"Method not allowed"

- Verify the form is submitting to the correct URL
- Check that PHP file has proper permissions
- Ensure web server is configured to execute PHP

CORS errors

- The PHP file includes CORS headers for cross-origin requests
- If still having issues, check your web server configuration

Form not submitting

- Check browser console for JavaScript errors
- Verify the fetch URL matches your domain structure
- Test the PHP endpoint directly with a tool like Postman

Debug Mode

Add this to the top of `submit-commission.php` for debugging:

PHP

```
error_reporting(E_ALL);  
ini_set('display_errors', 1);
```

Remove in production!

Email Notifications

The system can send email notifications when new commissions are submitted. Configure your server's mail system or use a service like:

- **SendGrid**
- **Mailgun**
- **Amazon SES**
- **SMTP relay**

Update the email configuration in `config.php` and modify the `mail()` function in `submit-commission.php` accordingly.

This setup provides a complete backend solution for handling prayer commission requests with proper database storage and email notifications.