CINEBOOK INSTALLATION GUIDE - SUPPLEMENTARY DOCUMENTATION

Additional Documentation for CineBook Installation

1. DETAILED SOFTWARE VERSION REQUIREMENTS

PHP Requirements

Software	Version	Notes
PHP	>= 8.2	Required
PHP GD Extension	Enabled	Required for QR Code
PHP OpenSSL	Enabled	Required
PHP PDO	Enabled	Required
PHP Mbstring	Enabled	Required
PHP Tokenizer	Enabled	Required
PHP XML	Enabled	Required
PHP Ctype	Enabled	Required
PHP JSON	Enabled	Required

Node.js Requirements

Software	Version	Notes
Node.js	>= 18.0	Required for Vite v7 and Tailwind v4
npm	>= 9.0	Bundled with Node.js

Checking PHP Extensions (XAMPP)

1. Open file `C:\xampp\php\php.ini`
2. Find and remove the `;` before these lines (if present):

```
extension=gd
extension=openssl
extension=pdo_mysql
extension=mbstring
```

3. Restart Apache in XAMPP Control Panel

2. CREATE DATABASE BEFORE IMPORTING

IMPORTANT: You must create the `cinebook` database before importing SQL files.

Method 1: Via phpMyAdmin

1. Open http://localhost/phpmyadmin
2. Click "New" in the left sidebar
3. Enter database name: `cinebook`
4. Select Collation: `utf8mb4_unicode_ci`
5. Click "Create"

Method 2: Via Terminal

```
mysql -u root -p -e "CREATE DATABASE cinebook CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;"
```

---

3. GENERATE APPLICATION KEY (IMPORTANT!)

After cloning the project and before running, you **MUST** generate the Application Key:

```
php artisan key:generate
```

If you skip this step:

- Sessions will not work
- Encryption will fail
- Application will throw errors

---

4. CREATE STORAGE SYMBOLIC LINK

To enable file uploads and display images (avatars, movie posters):

```
php artisan storage:link
```

This command creates a symlink from `public/storage` -> `storage/app/public`

---

5. EMAIL CONFIGURATION (OPTIONAL)

Option A: Using Log Driver (Development - Default)

Emails will be written to log files instead of being sent:

```
MAIL_MAILER=log
```

Option B: Using MailHog (Development)

1. Download MailHog: https://github.com/mailhog/MailHog/releases
2. Run MailHog.exe
3. Configure .env:

```
MAIL_MAILER=smtp
MAIL_HOST=127.0.0.1
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

4. View emails at: http://localhost:8025

Option C: Using Gmail SMTP (Production)

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=your-email@gmail.com
MAIL_PASSWORD=your-app-password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="noreply@cinebook.com"
MAIL_FROM_NAME="TCA Cine"
```

Note: You need to create an App Password in Google Account:

1. Go to https://myaccount.google.com/security
2. Enable 2-Step Verification
3. Create an App Password for "Mail"

---

6. QUEUE WORKER (FOR EMAIL JOBS)

CineBook uses database queue for sending emails. To make emails work:

Run Queue Worker

```
php artisan queue:listen --tries=1
```

Or use Composer Script (Recommended)

```
composer dev
```

This script runs simultaneously:

- Laravel Server (port 8000)
- Queue Worker
- Log Viewer (Pail)
- Vite Dev Server (port 5173)

---

7. SCHEDULED COMMANDS (CRON JOBS)

CineBook has automated commands:

Send Showtime Reminders

```
php artisan email:showtime-reminders --hours=2
```

Sends reminder emails 2 hours before showtime.

Send Review Requests

```
php artisan email:review-requests --hours=2
```

Sends review invitation emails 2 hours after showtime ends.

Test Email

```
php artisan email:test your-email@example.com
```

Setting up Cron Job (Production)

Add to crontab:

```
* * * * * cd /path-to-project && php artisan schedule:run >> /dev/null 2>&1
```

---

8. COMPLETE .ENV CONFIGURATION

```

```

```
# Application
APP_NAME=CineBook
APP_ENV=local
APP_KEY=base64:xxx  # Auto-generated by key:generate
APP_DEBUG=true
APP_URL=http://localhost:8000
APP_TIMEZONE=Asia/Ho_Chi_Minh

# Database
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=cinebook
DB_USERNAME=root
DB_PASSWORD=

# Session & Cache (using database)
SESSION_DRIVER=database
SESSION_LIFETIME=120
CACHE_STORE=database
QUEUE_CONNECTION=database

# Mail (choose one option from section 5)
MAIL_MAILER=log
```

---

9. CORRECT INSTALLATION ORDER

1. Clone repository
2. `composer install`
3. Copy `.env.example` -> `.env`
4. Configure `.env` (database, mail...)
5. `php artisan key:generate` **IMPORTANT**
6. Create database `cinebook` in phpMyAdmin
7. Import `mySQL/mySQL.sql` (schema)
8. Import `mySQL/data.sql` (sample data)
9. `php artisan storage:link`
10. `npm install`
11. `npm run dev` (terminal 1)
12. `php artisan serve` (terminal 2)
13. Access http://localhost:8000

---

10. PRODUCTION BUILD

When deploying to production:

```
# Build frontend assets
npm run build

# Optimize Laravel
php artisan config:cache
php artisan route:cache
php artisan view:cache

# Set APP_ENV in .env
APP_ENV=production
APP_DEBUG=false
```

---

11. QR CODE CHECK-IN SYSTEM

Browser Requirements

- Chrome 53+ / Firefox 36+ / Safari 11+
- HTTPS or localhost (to access camera)
- Grant camera permission to browser

Features

- Scan ticket QR codes at Admin > QR Check-in
- Each seat has its own QR code
- Couple seats have 2 QR codes

---

12. TROUBLESHOOTING - COMMON ERRORS

Error: "No application encryption key has been specified"

```
php artisan key:generate
```

Error: "SQLSTATE[HY000] [1049] Unknown database 'cinebook'"

Create database `cinebook` in phpMyAdmin before importing SQL.

Error: "The stream or file storage/logs/laravel.log could not be opened"

```
# Windows
icacls storage /grant Everyone:F /T
icacls bootstrap\cache /grant Everyone:F /T

# Linux/Mac
chmod -R 775 storage bootstrap/cache
```

Error: "Vite manifest not found"

```
npm run dev
# or
npm run build
```

Error: "Class 'Intervention\Image...' not found" (QR Code)

Check if PHP GD extension is enabled in php.ini

Error: "Mix manifest does not exist"

This project uses Vite, not Mix. Run `npm run dev`.

Emails not sending

1. Check MAIL_* configuration in .env
2. Run queue worker: `php artisan queue:listen`
3. Check logs: `storage/logs/laravel.log`

Images not displaying

```
php artisan storage:link
```

---

13. TEST ACCOUNTS (Detailed)

Role	Email	Password	Notes
Admin	admin@cinebook.com	123456	Full admin privileges
User	user1@gmail.com	123456	Test account
User	user2@gmail.com	123456	Test account
...	user3-15@gmail.com	123456	13 other accounts

---

14. IMPORTANT URLS

Page	URL
Homepage	http://localhost:8000
Admin Dashboard	http://localhost:8000/admin
QR Check-in	http://localhost:8000/admin/qr-checkin
phpMyAdmin	http://localhost/phpmyadmin

Page	URL
Vite Dev	http://localhost:5173
MailHog (if used)	http://localhost:8025

---

15. IMPORTANT DIRECTORY STRUCTURE

```
cinebook/
|-- app/
|   |-- Console/Commands/    # Scheduled commands
|   |-- Http/Controllers/    # Controllers
|   |-- Mail/                # Email templates
|   |-- Models/              # Eloquent models
|-- database/
|   |-- migrations/          # Database migrations
|   |-- seeders/             # Data seeders
|-- mySQL/
|   |-- mySQL.sql           # Database schema
|   |-- data.sql            # Sample data
|-- public/
|   |-- storage -> ../storage/app/public  # Symlink
|-- resources/
|   |-- css/                # Stylesheets
|   |-- js/                 # JavaScript
|   |-- views/              # Blade templates
|-- storage/
|   |-- app/public/         # Uploaded files
|-- .env                    # Environment config
|-- composer.json           # PHP dependencies
|-- package.json            # Node dependencies
```

---

16. DEVELOPMENT RECOMMENDATIONS

Use Composer Dev Script

```
composer dev
```

Runs all necessary services in one command.

Recommended VS Code Extensions

- Laravel Blade Snippets
- PHP Intelephense
- Tailwind CSS IntelliSense
- ES7+ React/Redux/React-Native snippets

Database GUI Tools

- phpMyAdmin (bundled with XAMPP)
- TablePlus (recommended)
- DBeaver (free)

Supplementary Documentation for CineBook Installation Guide Version 1.0 - January 2026