

# Deployment Instructions for Laravel/Vue Application

## ### Localhost Deployment

### 1. \*\*Install Prerequisites\*\*

Ensure the following tools are installed:

- **PHP** ( $\geq 8.0$ )
- **Composer**
- **Node.js** & **npm**
- **MySQL**

Install Laravel globally:

```
composer global require laravel/installer
```

### 2. \*\*Set Up the Project\*\*

Clone the project repository:

```
git clone https://github.com/tundeseun/accessibilityanalyzer.git
```

```
cd your-repository-folder
```

Install Laravel dependencies:

```
composer install
```

Install Vue and other frontend dependencies:

```
npm install
```

### 3. \*\*Environment Configuration\*\*

- Copy `.env.example` to `.env`:

```
cp .env.example .env
```

- **Generate an application key:**

**php artisan key:generate**

#### **4. \*\*Build Frontend Assets\*\***

- **Install and configure Tailwind CSS:**

**npm install -D tailwindcss postcss autoprefixer**

**npx tailwindcss init**

- **Compile Vue files and Tailwind assets:**

**npm run dev**

#### **5. \*\*Run the Application\*\***

**Start the Laravel development server:**

**php artisan serve**

**Visit the application at <http://127.0.0.1:8000>.**

---

### **### Docker Deployment (Optional)**

#### **1. \*\*Install Docker\*\***

**Install Docker and Docker Compose on your system.**

#### **2. \*\*Set Up Docker\*\***

**Create a `docker-compose.yml` file in the project root:**

```
``yaml
```

```
version: '3.8'
```

**services:**

**app:**

**build:**

**context:** .

**dockerfile:** Dockerfile

**ports:**

- "8000:8000"

**volumes:**

- ./var/www/html

**working\_dir:** /var/www/html

**environment:**

- DB\_HOST=mysql

- DB\_DATABASE=your\_database

- DB\_USERNAME=your\_username

- DB\_PASSWORD=your\_password

**mysql:**

**image:** mysql:8.0

**environment:**

**MYSQL\_ROOT\_PASSWORD:** root

**MYSQL\_DATABASE:** your\_database

**MYSQL\_USER:** your\_username

**MYSQL\_PASSWORD:** your\_password

...

### **3. \*\*Create Dockerfile\*\***

**Create a `Dockerfile` for Laravel:**

```
```dockerfile
FROM php:8.0-fpm

RUN apt-get update && apt-get install -y      libpng-dev      libjpeg-dev
libfreetype6-dev      zip      unzip      git      curl

RUN docker-php-ext-install pdo_mysql gd

COPY . /var/www/html

WORKDIR /var/www/html

RUN composer install

```
```

#### 4. **\*\*Start Docker Containers\*\***

```
docker-compose up -d
```

---

### ### Online Deployment

#### 1. **\*\*Prepare the Hosting Environment\*\***

- Ensure the server supports **\*\*PHP\*\*** ( $\geq 8.0$ ), **\*\*MySQL\*\***, and optionally **\*\*Node.js\*\*** for building Vue files.

#### 2. **\*\*Set Up the Project\*\***

- Clone the repository or upload project files using SSH or FTP.
- SSH into the server and navigate to the project directory:  

```
cd /path/to/project
```

#### 3. **\*\*Install Dependencies\*\***

**composer install --optimize-autoloader --no-dev**

**npm install**

**npm run build**

#### **4. \*\*Environment Configuration\*\***

**- Copy `.env.example` to `.env`:**

**cp .env.example .env**

**- Update `.env` with server-specific details and generate an app key:**

**php artisan key:generate**

#### **5. \*\*Database Setup\*\***

**- Create a database and update credentials in `.env`.**

**- Run migrations and seed the database:**

**php artisan migrate --force**

#### **6. \*\*Set Permissions\*\***

**chmod -R 775 storage bootstrap/cache**

**chown -R www-data:www-data storage bootstrap/cache**

#### **7. \*\*Set Up Web Server\*\***

**Configure \*\*Apache\*\* or \*\*Nginx\*\* to point to the public directory of the project.**

#### **8. \*\*Optimize Application\*\***

**Optimize for production:**

**php artisan optimize**

**php artisan config:cache**

**php artisan route:cache**

**php artisan view:cache**

## **9. \*\*Access the Application\*\***

**Open the domain in a web browser.**

---

### **### Additional Notes**

- Ensure `.env` contains all sensitive configurations.**
- Use \*\*Tailwind CSS\*\* to customize frontend styles.**
- Use \*\*Docker\*\* for consistent environment setups.**
- Regularly monitor logs at `storage/logs/laravel.log`.**