

# **DEPLOYMENT INSTRUCTIONS FOR LARAVEL/VUE APPLICATION ON LOCALHOST AND ONLINE**

## **Localhost Deployment**

### **1. Install Prerequisites:**

- Ensure the following are installed on your local machine:
  - PHP (>=8.0)
  - Composer
  - Node.js & npm
  - MySQL
- Install Laravel globally (if not already):

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 dependencies:

npm install

### **3. Environment Configuration:**

- Copy .env.example to .env:

cp .env.example .env

### **4. Build Frontend Assets:**

- Compile Vue files:

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>.

## **Online Deployment**

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

- Ensure the server supports:
  - PHP ( $\geq 8.0$ )
  - MySQL
  - Node.js (optional, for building Vue files)
- Set up a domain/subdomain.

### **2. Upload the Project:**

- Clone the repository or upload the project files to the server using SSH or FTP.

### **3. Install Dependencies:**

- SSH into the server and navigate to the project directory:
- `cd /path/to/project`

Install Laravel dependencies:

```
composer install --optimize-autoloader --no-dev
```

Install Vue dependencies and build production assets:

```
npm install
```

```
npm run build
```

### **4. Environment Configuration:**

- Copy `.env.example` to `.env`.
- Update the `.env` file with the server's database credentials and other configurations.

### **5. Database Setup:**

- Create a database on the server.
- Run migrations:
  - `php artisan migrate --force`

### **6. Set Permissions:**

- Ensure the storage and bootstrap/cache directories are writable
  - `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.
- Example Nginx configuration:

```
server {

    listen 80;

    server_name your-domain.com;

    root /path/to/project/public;

    index index.php index.html;


    location / {

        try_files $uri $uri/ /index.php?$query_string;

    }


    location ~ \.php$ {

        include snippets/fastcgi-php.conf;

        fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

        include fastcgi_params;

    }

}
```

## 8. Optimize Application:

- Optimize the application 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.

**Notes**

- Always use .env for sensitive configurations.
- Monitor logs (storage/logs/laravel.log) for debugging issues.