

Setup Guide - How to Run This Project on Any PC

 **IMPORTANT:** Don't Use `nuurem.sql` Directly!

Problem with SQL dumps:

- ✗ Database name hardcoded (`nuurem`)
- ✗ Missing `remember_token` column (needed for "Remember Me")
- ✗ Missing `admins` table (needed for admin login)
- ✗ Manual import required

Laravel solves this with Migrations!

 Complete Setup Instructions (For Team Members)

Step 1: Install Prerequisites

Make sure you have:

- PHP 8.2+ (`php --version`)
- Composer (`composer --version`)
- Node.js & npm (`node --version, npm --version`)
- MySQL (XAMPP or Laragon)

Step 2: Clone the Repository

```
cd "D:\Nsu courses\cse327project"
git clone https://github.com/Nasim227/CSE327_Group-6.git
cd CSE327_Group-6
```

Step 3: Install Dependencies

```
# Install PHP dependencies
composer install

# Install JavaScript dependencies
npm install
```

Step 4: Setup Environment File

```
# Copy the example environment file
copy .env.example .env
```

```
# Generate application key  
php artisan key:generate
```

Step 5: Configure Database

Edit `.env` file (use Notepad or VS Code):

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=cse327_app          # ← Change to YOUR database name  
DB_USERNAME=root                # ← Change if needed  
DB_PASSWORD=                      # ← Add password if you have one
```

Important: Each person can use a different database name! Laravel doesn't care.

Step 6: Create Database

Option A: Using phpMyAdmin

1. Open <http://localhost/phpmyadmin>
2. Click "New" on left sidebar
3. Create database named `cse327_app` (or whatever you put in `.env`)
4. Collation: `utf8mb4_general_ci`

Option B: Using Command Line

```
mysql -u root -p  
CREATE DATABASE cse327_app;  
exit;
```

Step 7: Run Migrations (CRITICAL STEP!)

```
php artisan migrate
```

What this does:

- Creates ALL tables automatically
- Including `user` table with `remember_token`
- Including `admins` table
- Sets up all columns correctly
- Works on ANY database name

Expected output:

```
Migration table created successfully.  
Migrating: 2025_11_19_170953_create_user_table  
Migrated: 2025_11_19_170953_create_user_table  
Migrating: 2025_12_03_222000_add_remember_token_to_user_table  
Migrated: 2025_12_03_222000_add_remember_token_to_user_table  
Migrating: 2025_12_03_230000_create_admins_table  
Migrated: 2025_12_03_230000_create_admins_table
```

Step 8: Build Frontend Assets

```
npm run build
```

Step 9: Start the Server

```
php artisan serve
```

Open browser: <http://localhost:8000>

✓ Verification Checklist

After setup, verify everything works:

- Can access <http://localhost:8000>
- Registration page loads: <http://localhost:8000/register>
- Can create a new user account
- Can login with created account
- "Remember Me" checkbox works
- Can logout
- Admin login page loads: <http://localhost:8000/admin/login>
- Admin registration works

📝 Run Tests (Optional but Recommended)

```
php artisan test
```

Expected result: All 41 tests pass

🔧 Common Issues & Solutions

Issue 1: "SQLSTATE[HY000] [1045] Access denied"

Solution: Wrong database credentials in `.env`

Fix:

```
DB_USERNAME=root      # Change to your MySQL username  
DB_PASSWORD=yourpass  # Add your MySQL password
```

Then run:

```
php artisan config:clear
```

Issue 2: "Base table or view not found: 'user'"

Solution: Forgot to run migrations

Fix:

```
php artisan migrate
```

Issue 3: "Remember Me doesn't work"

Solution: Missing `remember_token` column

Fix (run migration again):

```
php artisan migrate:fresh
```

Issue 4: "Class 'Admin' not found"

Solution: Missing composer autoload

Fix:

```
composer dump-autoload
```

Issue 5: Port 8000 already in use

Solution: Use a different port

```
php artisan serve --port=8001
```

Then visit: <http://localhost:8001>

📊 What Migrations Create

Migrations automatically create these tables:

1. `user` table

```
User_id (INT, PRIMARY KEY, AUTO_INCREMENT)
First_name (VARCHAR 200)
Last_name (VARCHAR 200)
Email (VARCHAR 200, UNIQUE)
Password (VARCHAR 255)
remember_token (VARCHAR 100, NULLABLE) ← For "Remember Me"
```

2. `admins` table

```
id (INT, PRIMARY KEY, AUTO_INCREMENT)
name (VARCHAR 255)
email (VARCHAR 255, UNIQUE)
password (VARCHAR 255)
remember_token (VARCHAR 100, NULLABLE)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)
```

3. Other tables from `nuurem.sql`

- cart
 - cart_item
 - category
 - orders
 - order_items
 - products
-

⌚ Updating the Project (If Code Changes)

When someone updates the code and you pull changes:

```
# Get latest code
git pull origin Sadman

# Update dependencies (if composer.json changed)
composer install

# Update JavaScript dependencies (if package.json changed)
```

```
npm install

# Run any new migrations (if database changed)
php artisan migrate

# Rebuild frontend (if views/CSS changed)
npm run build
```

⌚ Why Migrations > SQL Dumps

SQL Dump (<code>nuurem.sql</code>)	Laravel Migrations
✗ Hardcoded database name	<input checked="" type="checkbox"/> Works with ANY database name
✗ Must manually import	<input checked="" type="checkbox"/> Automatic with <code>php artisan migrate</code>
✗ Conflicts if database exists	<input checked="" type="checkbox"/> Only adds what's missing
✗ Hard to track changes	<input checked="" type="checkbox"/> Git tracks all changes
✗ Missing new columns	<input checked="" type="checkbox"/> Always up-to-date

📝 For Presentation Demo

If you need to demo on instructor's PC:

1. Quick Setup (5 minutes):

```
composer install
npm install
copy .env.example .env
php artisan key:generate
# Edit .env with database name
php artisan migrate
npm run build
php artisan serve
```

2. Show that it works:

- Register new account
- Login
- Test "Remember Me"
- Show admin login

3. If asked "How does database work?":

"We use Laravel migrations. Each team member runs `php artisan migrate` which creates the exact same table structure on any database name they choose. This is better than SQL dumps because it's automatic, version-controlled, and prevents conflicts."

Emergency Reset

If everything breaks, complete reset:

```
# Clear all caches  
php artisan cache:clear  
php artisan config:clear  
php artisan route:clear  
php artisan view:clear  
  
# Drop all tables and recreate  
php artisan migrate:fresh  
  
# Regenerate autoload  
composer dump-autoload  
  
# Rebuild frontend  
npm run build
```

Getting Help

If setup fails:

1. Check `.env` file database credentials
 2. Verify MySQL is running (start XAMPP/Laragon)
 3. Run `php artisan migrate` again
 4. Check `storage/logs/laravel.log` for errors
 5. Ask team member who has it working
-

Pro Tip: Database Seeders (Optional)

To add test data automatically:

```
php artisan db:seed
```

This creates dummy users/products for testing.

Created: 2025-12-04

Author: Sadman (CSE327 Group-6)