

Next.js + MySQL

This is a [Next.js](#) project that uses [Prisma](#) to connect to a [PlanetScale](#) MySQL database and [Tailwind CSS](#) for styling.

Demo

<https://next-mysql.vercel.app>

Prerequisites

- [Node.js](#)
- [PlanetScale CLI](#)
- Authenticate the CLI with the following command:

```
pscale auth login
```

Set up the database

Create a new database with the following command:

```
pscale database create <DATABASE_NAME>
```

A branch, `main`, was automatically created when you created your database, so you can use that for `BRANCH_NAME` in the steps below.

Set up the starter Next.js app

Execute [create-next-app](#) with [npm](#) or [Yarn](#) to bootstrap the example:

```
npx create-next-app --example with-mysql nextjs-mysql
# or
yarn create next-app --example with-mysql nextjs-mysql
# or
pnpm create next-app -- --example with-mysql nextjs-mysql
```

Next, you'll need to create a database username and password through the CLI to connect to your application. If you'd prefer to use the dashboard for this step, you can find those instructions in the [Connection Strings documentation](#) and then come back here to finish setup.

First, create your `.env` file by renaming the `.env.example` file to `.env`:

```
mv .env.example .env
```

Next, using the PlanetScale CLI, create a new username and password for the branch of your database:

```
pscale password create <DATABASE_NAME> <BRANCH_NAME> <PASSWORD_NAME>
```

The `PASSWORD_NAME` value represents the name of the username and password being generated. You can have multiple credentials for a branch, so this gives you a way to categorize them. To manage your passwords in the dashboard, go to your database overview page, click "Settings", and then click "Passwords".

Take note of the values returned to you, as you won't be able to see this password again.

```
Password production-password was successfully created.
Please save the values below as they will not be shown again
```

NAME	USERNAME	ACCESS HOST URL	ROLE
PLAIN TEXT			
-----	-----	-----	-----
production-password	xxxxxxxxxxxxxx	xxxxxx.us-east-2.pscale.cloud	Can Read & Write
pscale_pw_xxxxxxx			

You'll use these properties to construct your connection string, which will be the value for `DATABASE_URL` in your `.env` file. Update the `DATABASE_URL` property with your connection string in the following format:

```
mysql://<USERNAME>:<PLAIN_TEXT_PASSWORD>@<ACCESS_HOST_URL>/<DATABASE_NAME>?
sslaccept=strict
```

Push the database schema to your PlanetScale database using Prisma.

```
npx prisma db push
```

Run the seed script to populate your database with `Product` and `Category` data.

```
npm run seed
```

Run the App

Run the app with following command:

```
npm run dev
```

Open your browser at localhost:3000 to see the running application.

Deploy your own

After you've got your application running locally, it's time to deploy it. To do so, you'll need to promote your database branch (`main` by default) to be the production branch ([read the branching documentation for more information](#)).

```
pscale branch promote <DATABASE_NAME> <BRANCH_NAME>
```

Now that your branch has been promoted to production, you can either use the existing password you generated earlier for running locally or create a new password. Regardless, you'll need a password in the deployment steps below.

Deploy the example using [Vercel](#):



Make sure to update the `DATABASE_URL` variable during this setup process.