

Kurless Store

Website: <https://kurless-acm.netlify.app/emailsub>

Table of Contents

1. Project Overview
 2. Features
 3. Getting Started
 - Prerequisites
 - Installation
 - Environment Variables
 4. Project Structure
 5. Usage
 - Accessing the Website
 - Updating the Website
 - Authentication
 - Subscription
 6. Scripts
 7. Deployment
 8. Troubleshooting
 9. Contributing
 10. License
 11. Contact Information
-

1. Project Overview

Kurless Store is a responsive e-commerce website built with HTML, CSS, and JavaScript. It integrates with Shopify for product management and Netlify for seamless deployment and serverless functions. The website features user authentication, a product catalog, subscription form, and contact form, providing a comprehensive online shopping experience.

2. Features

- **Responsive Design:** Optimized for various devices and screen sizes.
 - **Navigation Bar:** Easily navigate between Home, Shop, Catalogue, and Contact Us pages.
 - **Authentication:** Secure login system to protect specific sections of the website.
 - **Product Catalog:** Display of popular and all products with integration to Shopify Buy Buttons.
 - **Subscription Form:** Users can subscribe to stay updated via email.
 - **Contact Form:** Users can reach out through a contact form.
 - **Notification Bar:** Persistent notification about free shipping offers.
 - **Scroll-to-Top Button:** Convenient button to quickly return to the top of the page.
 - **Serverless Functions:** Secure password verification using Netlify Functions.
 - **Environment Variables:** Secure handling of sensitive information like password hashes.
-

3. Getting Started

Prerequisites

Before you begin, ensure you have met the following requirements:

- **Git:** Installed on your local machine. [Download Git](#)
- **Node.js & npm:** Installed for managing serverless functions. [Download Node.js](#)
- **Netlify Account:** For deploying the website and managing serverless functions. [Sign Up for Netlify](#)
- **Shopify Store:** For managing products and generating Buy Buttons. [Shopify Setup](#)

Installation

Clone the Repository:

bash

Copy code

```
git clone https://github.com/yourusername/kurless-store.git
cd kurless-store
```

1.

Navigate to the Functions Directory:

bash

Copy code

```
cd netlify/functions
```

2.

Install Dependencies:

bash

Copy code

```
npm install
```

Environment Variables

1. Create a `.env` File:

In the root directory of your project, create a file named `.env`.

Add the Following Variable:

env

Copy code

```
UNIVERSAL_PASSWORD_HASH=$2b$10$0VTWltzRXsUQMPusaxSGqePOK2TgH4hHSC5Xa34
ljw/5itbvs1ZQe
```

2.

- Replace the value with your actual hashed password. To generate a new hash, use the provided `hashPassword.js` script.

Ensure `.env` is Ignored:

Add `.env` to your `.gitignore` to prevent it from being committed to version control.

bash

Copy code

```
# .gitignore
.env
```

4. Project Structure

```
scss
Copy code
kurless-store/
├── assets/
│   ├── images/
│   │   ├── logo.png
│   │   ├── socks.png
│   │   ├── Sculpt2.png
│   │   ├── Sculpt3.png
│   │   └── ... (other images)
│   ├── css/
│   │   └── styles.css
│   ├── js/
│   │   ├── scripts.js
│   │   ├── netlify/
│   │   └── functions/
│   ├── verifyPassword.js
│   └── package.json
├── index.html
├── login.html
├── shop.html
├── about.html
├── contact.html
├── emailsub.html
├── hashPassword.js
├── README.md
└── ... (other files)
```

5. Usage

Accessing the Website

Visit the live website at: <https://kurless-acm.netlify.app/emailsub>

Updating the Website

1. Make Changes Locally:

Edit the HTML, CSS, or JavaScript files as needed.

Commit Your Changes:

```
bash
Copy code
git add .
git commit -m "Your descriptive commit message"
```

2.

Push to Remote Repository:

```
bash
Copy code
git push origin main
```

3.

- Netlify will automatically detect the changes, rebuild, and redeploy the website.

Authentication

The website includes a secure login system to protect certain pages.

1. Login Page:

Navigate to the login page at `/login.html`.

2. Entering Password:

Enter the password to gain access. The password is verified using a serverless function.

3. Serverless Function:

The `verifyPassword.js` function handles password verification using bcrypt. It sets an `auth_token` cookie upon successful authentication.

4. Protected Pages:

Ensure that pages requiring authentication check for the `auth_token` cookie to grant access.

Subscription

Users can subscribe to updates via the subscription form.

1. Subscription Form:

Located on `/emailsub.html`.

2. Form Handling:

The form submits to Formspree (<https://formspree.io/f/mrbgpdyy>) for handling email subscriptions.

3. Success/Error Messages:

Users receive feedback upon successful subscription or errors.

6. Scripts

Generating a Hashed Password

To create a new hashed password for authentication:

Navigate to the Project Root:

bash

Copy code

```
cd kurless-store
```

1.

Run the Hash Password Script:

bash

Copy code

```
node hashPassword.js
```

2.

Output:

The script will output a hashed password. Copy this hash and update the

`UNIVERSAL_PASSWORD_HASH` in your `.env` file.

javascript

Copy code

```
Hashed Password: $2b$10$...
```

7. Deployment

The website is deployed using Netlify, which handles continuous deployment from the connected Git repository.

1. Connect Repository to Netlify:

- Ensure your Git repository is connected to Netlify. This allows automatic deployments on push.

2. Set Environment Variables in Netlify:

- In the Netlify dashboard, navigate to your site settings.
- Go to **Build & Deploy > Environment > Environment Variables**.
- Add `UNIVERSAL_PASSWORD_HASH` with your hashed password.

3. Deploy Functions:

- Ensure that the `netlify/functions` directory contains your serverless functions.
- Netlify will automatically deploy these functions upon pushing changes.

4. Monitor Deployments:

- Use the Netlify dashboard to monitor build and deployment status.
-

8. Troubleshooting

• Build Failures:

- Check the Netlify build logs for error messages.
- Ensure all dependencies are installed correctly in the `functions` directory.

• Authentication Issues:

- Verify that the `UNIVERSAL_PASSWORD_HASH` is correctly set in environment variables.
 - Ensure the serverless function is deployed and accessible.

• Form Submissions Not Working:

- Check the Formspree endpoint in `emailsub.html`.
- Ensure that Formspree is correctly configured to handle submissions.

• Styling or Layout Issues:

- Clear your browser cache and perform a hard refresh (`Ctrl + F5` or `Cmd + Shift + R`).
 - Verify that all CSS files are correctly linked.
-

9. License

This project is licensed under the MIT License.

10. Contact Information

For any inquiries or support, please contact me at cristian.simut@yahoo.com

Note: Replace placeholder URLs like <https://github.com/yourusername/kurless-store.git> with your actual repository URL and ensure all links (e.g., contact email) are correctly pointed to your resources.