TikTok Rescue

TikTok Rescue is a comprehensive web-based solution designed to help users securely download, preserve, and transfer their TikTok content before potential disruptions. The service offers a range of features from bulk downloading, content migration to YouTube, and affiliate program integration. It is built using HTML, CSS, JavaScript, Firebase, and TailwindCSS, ensuring a responsive and secure user experience across various devices.

Table of Contents

Features

- **Bulk Download**: Download individual or multiple TikTok videos simultaneously.
- **Content Transfer**: Migrate videos to platforms like YouTube and Instagram.
- **Secure Authentication**: Use Google, TikTok, or email sign-in methods powered by Firebase.
- **Affiliate Program**: Join an affiliate program to earn commissions through referrals.
- **Responsive Design**: User-friendly interface built with TailwindCSS that adapts to all screen sizes.
- **Real-time Support**: Contact form for user inquiries integrated with email notifications.

Installation

1. **Clone the repository**:

```
git clone https://github.com/yourusername/tiktok-rescue.git
```

2. **Navigate to the project directory**:

```
cd tiktok-rescue
```

3. **Install dependencies** (if applicable):
    - For Firebase functions:

```
cd functions
npm install
```

4. **Configure Firebase**:
    - Update the `firebaseConfig` in `auth.js` with your Firebase project credentials.
    - Deploy Firebase functions using:

```
firebase deploy --only functions
```

Usage

- Open `index.html` in a web browser to access the landing page.
- Use navigation links to explore different sections: Home, Features, Downloads, FAQ, Affiliates, etc.
- Sign up or sign in using your preferred method on the authentication page.
- For downloading content or migrating videos, follow on-screen instructions provided after authentication.
- Contact support via the Contact Us page to submit any questions or feedback.

Project Structure

```
tiktok-rescue/
├── index.html
├── privacy-policy.html
├── send_mail.php
├── terms-of-service.html
├── contact_us.html
├── affiliates.html
├── paypal-service.html
├── dashboard.html
├── auth.html
├── download.html
├── about.html
├── header.html
├── footer.html
├── base-template.html
├── styles.css
├── scripts.js
├── auth.js
├── index.js                # Firebase function for email
notifications
├── contact_error.html
├── contact_success.html
└── README.md
```

- **HTML Files**: Contain the structure and content for various pages including Home, Privacy Policy, Terms of Service, Contact Us, Affiliates, etc.
- **CSS File**: `styles.css` defines custom styling rules and button behaviors.
- **JavaScript Files**:
  - `scripts.js`: Handles dynamic loading of header/footer and mobile menu functionality.
  - `auth.js`: Manages user authentication with Firebase, including sign-in, sign-up, and state management.
  - `index.js`: Contains Firebase Cloud Function to send email notifications on new contact form submissions.
- **Template Files**: `base-template.html` provides a reusable structure for consistent page layouts.

Features of the Website:

- **Bulk Download**: Users can download multiple TikTok videos at once.
- **Content Transfer**: Allows transferring videos to platforms like YouTube or Instagram.
- **Secure Authentication**: Sign in/up with Google, TikTok, or email using Firebase, including UI toggles between Sign Up and Sign In.
- **Responsive Design**: Built with TailwindCSS for a mobile-friendly and responsive interface.
- **Affiliate Program**: Provides information on joining the affiliate program, requirements, benefits, and earnings.
- **Real-time Support and Contact**: Includes a contact form that sends email notifications using Firebase functions and Nodemailer/PHPMailer.
- **User Dashboard**: Displays personalized information, profile data, and user-specific content after authentication.
- **FAQ Section**: Contains frequently asked questions with answers about the service, installation, usage, security, and more.
- **PayPal Fees Guide**: Detailed explanation of PayPal fee calculations and optimal transfer amounts for affiliates and users.

Page-by-Page Breakdown:

1. **index.html (Home Page)**
   - Hero section introducing the service with a call-to-action button for downloads.
   - Pricing section detailing one-time payment features.
   - Video demonstration embedded via YouTube iframe.
   - "How It Works" section with steps to secure TikTok content.
   - Feature highlights and benefits of the service.
   - Affiliate program teaser inviting users to join.
   - FAQ snippet and a download invitation for free trials.
2. **privacy-policy.html**

- Outlines the privacy practices, data collection methods, usage of personal, usage, and content data.
- Details how user information is shared, data security measures, user rights, and handling children's privacy.
- Information on international data transfers and policy changes.

3. **terms-of-service.html**
   - Defines Terms and Conditions for using the service.
   - Includes user eligibility, permitted/prohibited uses, and compliance with third-party platforms.
   - Details on affiliate program participation, payment, refunds, intellectual property, disclaimer, liability, indemnification, termination, and governing law.
   - Describes user rights, responsibilities, and obligations.
4. **footer.html**
   - Contains navigational links to Privacy Policy, Terms of Service, Contact Us, and copyright.
   - Displays company credits and links to WebAlly Software Development.
5. **header.html**
   - Navigation bar with links to Home, Features, Download, How It Works, FAQ, and Affiliates sections.
   - Responsive design elements including mobile hamburger menu.
6. **base-template.html**
   - A template structure that includes header and footer placeholders.
   - Provides consistent styling and layout for various pages.
7. **contact_us.html**
   - Contact form for users to send messages, including fields for name, email, and message.
   - Submits data to `send_email.php` for processing.
   - Provides user feedback on submission status.
8. **styles.css**
   - Contains custom CSS styles for buttons and hover effects used across the website.
9. **scripts.js**
   - JavaScript handling dynamic loading of header/footer.
   - Manages mobile navigation menu interactivity and event listeners for menu toggling.
10. **auth.html**
    - Authentication page allowing users to sign in or sign up.
    - Provides buttons for Google sign-in and email-based authentication.

- Contains toggling functionality between sign-in and sign-up modes.

11. **download.html**
    - Page intended for download functionality (currently serves as a template with placeholders for unique content).
12. **about.html**
    - Placeholder page for information about TikTok Rescue or related content (structure similar to base template).
13. **dashboard.html**
    - User dashboard accessible after authentication.
    - Displays a welcome message, user profile information, and user-specific content sections.
    - Provides a personalized experience for logged-in users.
14. **affiliates.html**
    - Dedicated page for the affiliate program.
    - Includes a hero section inviting users to join, benefits grid, requirements for affiliates, and instructions on how the program works.
    - FAQ section addressing common affiliate questions.
    - Call-to-action for signing up and earning commissions.
15. **paypal-service.html**
    - Guide on understanding PayPal transaction fees.
    - Breaks down fee structures, analyzes break-even points, optimal transfer amounts, and least viable transfer amounts.
    - Provides examples and visual breakdowns for different transfer scenarios.
16. **index.js**
    - Firebase Cloud Function script.
    - Listens for new messages in Firestore and sends email notifications using Nodemailer.
17. **contact_error.html**
    - Error page displayed when a contact form submission fails.
    - Provides a friendly error message and a link back to the Contact Us page.
18. **contact_success.html**
    - Success page displayed after a successful contact form submission.
    - Thanks the user for their message and provides a link back to the home page.

Additional File Provided: send_email.php

- **send_email.php**
    - Handles form submissions from `contact_us.html`.
    - Utilizes PHPMailer to send an email with the submitted contact form data.

- Sanitizes user input to prevent security issues.
- Configured to use IONOS SMTP server for sending emails.
- On success: redirects user to `contact_success.html`.
- On failure: redirects user to `contact_error.html`.
- Provides a fallback message if accessed directly via non-POST methods.

Authentication:

The authentication system for TikTok Rescue is built on top of Firebase Authentication and Firestore, providing secure and versatile sign-in/up mechanisms. It supports Google sign-in as well as email/password-based authentication, and integrates with a dynamic UI that toggles between sign-up and sign-in modes. Here are the key components and flow:

1. **Firebase Initialization:**
   - The system initializes Firebase using a configuration object in `auth.js`. It imports Firebase SDK modules (`firebase-app`, `firebase-auth`, and `firebase-firestore`) and sets up connections to Firebase services.
2. **UI Rendering and Toggles:**
   - The authentication UI is dynamically generated within the `initializeAuthUI()` function in `auth.js`. This function creates HTML for Google sign-in, email/password fields, and a button that dynamically changes text based on whether the user is signing up or signing in.
   - A toggle mechanism (`toggleAuthMode()`) allows users to switch between Sign Up and Sign In forms. The UI updates accordingly without needing page reloads.
3. **Google Sign-In:**
   - The `signInWithGoogle()` function creates a new Google Auth Provider instance and uses `signInWithPopup()` to authenticate the user through a Google account.
   - Upon successful authentication, it checks if the user is new. If so, it calls `createUserProfile()` to store initial user data in Firestore.
   - After authentication, the user is redirected to the dashboard and their session information is stored in session storage.
4. **Email/Password Authentication:**
   - The `handleEmailAuth()` function handles both sign-up and sign-in using email and password.
   - For sign-up, it uses `createUserWithEmailAndPassword()`, creates a Firestore user profile via `createUserProfile()`, and then follows a similar post-authentication process.
   - For sign-in, it uses `signInWithEmailAndPassword()` to authenticate existing users.

- Successful authentication leads to redirection and session management similar to Google sign-in.

5. **User Profile Management:**
   - When a new user is created, `createUserProfile()` stores their email, display name, photo URL, and timestamps for account creation and last login in a Firestore document under the `users` collection.
   - After every successful authentication, the system updates the user's `lastLogin` timestamp in Firestore.
6. **Auth State Listener:**
   - The system sets up an authentication state listener with `auth.onAuthStateChanged(handleAuthStateChange)`. This function executes whenever the user's sign-in state changes.
   - If the user is signed in, it retrieves their profile from Firestore and updates the UI accordingly using `updateUIForAuthenticatedUser()`.
   - If the user is signed out, it resets UI elements with `updateUIForUnauthenticatedUser()`.
7. **Sign Out:**
   - The `signOut()` function calls Firebase's `auth.signOut()` method, clears the session storage, and redirects the user back to the home page.

Security and Best Practices:

- Inputs are sanitized during sign-up and sign-in to prevent injection attacks.
- Firebase securely handles password storage and authentication.
- Sensitive operations, like creating user profiles and updating login timestamps, are performed on the server-side using secure Firestore calls.

Overall, the auth system is designed to be robust, user-friendly, and secure, leveraging Firebase services to handle the heavy lifting of authentication and data storage while providing a smooth UI experience.

Contributing

Contributions are welcome! Please fork the repository and submit a pull request for any improvements, bug fixes, or new features.

License

This project is licensed under the MIT License. See the LICENSE file for details.