**MS VBA Password Remover Script by Python**

**Overview:**

This web-based tool is designed to allow users to remove the password protection from VBA (Visual Basic for Applications) code embedded within Microsoft Office documents, such as Excel, Word, and PowerPoint. It works with Office files from 2007 and later, typically saved in .xlsx, .docx, .pptx, and similar formats. The tool decrypts the VBA code embedded within these documents, making it accessible for users to edit or view.

The tool provides a user-friendly interface with both **light mode** and **dark mode** for flexibility, and it includes a **drag-and-drop file upload feature** to enhance user experience.

**Features:**

1. **Drag-and-Drop Upload**: Users can upload their Office files by dragging them into the web interface or selecting them through a file input dialog.
2. **Dark/Light Mode Toggle**: Users can toggle between a dark or light theme for better readability and comfort.
3. **VBA Decryption**: The main feature of the tool removes password protection from the VBA code embedded in Office files.
4. **Instructions**: The page includes step-by-step instructions for the user to follow, detailing how to remove the password from the VBA code manually after the document is decrypted.

**How it Works:**

1. **User Uploads File**:
   * The user uploads their Office document (.docx, .xlsx, or .pptx) either by dragging it into the designated area or by clicking to select it.
2. **Decryption Process**:
   * The backend Python script receives the uploaded file, extracts the VBA project from the Office file, and removes the password protection.
3. **Download Process**:
   * After the decryption is complete, the user is prompted to download a new version of the Office document without the password protection on the VBA code.
4. **Manual Steps for Removing VBA Password**:
   * After downloading the document, the user follows the instructions provided to set and later removes the VBA password protection within the Office application.

**Required Tools/Technologies:**

* **Backend**: Python (Flask for web framework, zipfile and openpyxl libraries for working with Office documents)
* **Frontend**: HTML, CSS, JavaScript (for user interface, including file upload and dark/light mode toggle)
* **File Formats Supported**: .docx, .xlsx, .pptx (Office 2007 and later)

**Frontend (HTML + CSS + JavaScript):**

The frontend consists of:

1. **File Upload Area**: A drag-and-drop box or file input area that allows users to upload Office documents.
2. **Dark Mode/Light Mode Toggle**: A button to switch between light and dark themes for better accessibility.
3. **Instructions**: A list of detailed instructions that guide the user through the manual process required to fully remove the VBA password.

**Key Files:**

1. **HTML** (index.html):
   * Contains the structure and content of the web page.
   * Provides an upload interface for Office files.
   * Includes the mode toggle button and instructions.
   * The JavaScript embedded in this file handles the dark/light mode toggle and drag-and-drop functionality.
2. **CSS** (styles.css):
   * Provides the styling for the page, including design for light/dark modes and the drag-and-drop area.
   * Ensures the webpage is responsive and looks good on various screen sizes.
3. **JavaScript** (inside the HTML file):
   * Manages the drag-and-drop file upload.
   * Handles the toggling of dark/light modes.

**Backend (Python Script):**

1. **Flask Web Server**:
   * The Flask server handles the request to upload an Office file and process it by removing the VBA password protection.
   * The uploaded file is processed using Python libraries such as zipfile to extract the contents of the Office file, and openpyxl (for .xlsx files) to manipulate the VBA project.
2. **VBA Password Removal Process**:
   * The script extracts the VBA project from the Office file and removes the password protection by modifying the underlying zip structure of the Office file.
   * After the process is complete, a new Office file is generated, and the user is given the option to download it.

**Instructions for Use:**

1. **Upload Your Office Document**:
   * Click on the "Drag & drop your file here, or click to select" area to choose an Office file (.docx, .xlsx, .pptx) from your computer, or drag and drop the file directly into the area.
2. **Confirm the Download**:
   * After the file is uploaded and processed, the system will provide a download link for the new document without the VBA password protection.
3. **Open the Downloaded Document**:
   * Open the downloaded document in your Office application (Word, Excel, PowerPoint).
   * Press ALT + F11 to open the VBA editor. You may see error messages about missing code or protection. Confirm these messages.
4. **Set a New Password**:
   * In the VBA editor window, do not expand the project.
   * Navigate to "Tools > VBA Project Properties" and go to the "Protection" tab.
   * Set a new password of your choice and leave the checkbox selected.
   * Save the document and close your Office application entirely.
5. **Remove the Password**:
   * Open the document again and press ALT + F11 to access the VBA editor.
   * Navigate to "Tools > VBA Project Properties" and go to the "Protection" tab.
   * Clear the checkbox and the password fields.
   * Save the document again.
6. **Password Removed**:
   * The password protection is now completely removed, and you can freely view and edit the VBA code as needed.

**Troubleshooting:**

* **Error Message During Decryption**:
  + If you see an error after uploading the file, ensure that the file is in a valid .docx, .xlsx, or .pptx format.
  + Verify that the document contains a protected VBA project.
* **Unable to Open the File After Decryption**:
  + If you cannot open the downloaded file or if it’s corrupted, try re-uploading the file and processing it again.
  + Make sure your Office application is up to date and supports the document format.
* **VBA Project Not Visible After Removal**:
  + If the VBA project still appears locked, ensure that the password was removed following the instructions in the manual steps.

**System Requirements:**

* **Python Version**: Python 3.6 or later.
* **Flask**: Web framework to run the web application.
* **Libraries**:
  + zipfile: For extracting and manipulating Office files (in zip format).
  + openpyxl: To work with .xlsx files and extract embedded VBA projects.
  + io: For handling file streams.

**How to Deploy Locally:**

1. **Install Python**:
   * Download and install Python 3.6 or later from [python.org](https://www.python.org/downloads/).
2. **Install Required Libraries**: Open a terminal or command prompt and install the required Python libraries:

pip install flask zipfile36 openpyxl

1. **Run the Flask Application**:
   * Save the Python script (app.py) and the HTML (index.html) and CSS (styles.css) files in a directory.
   * Navigate to that directory and run the Flask app using:

python app.py

1. **Access the Web Application**:
   * Once the Flask server is running, open your web browser and navigate to http://127.0.0.1:5000/ to use the tool locally.

**Security and Privacy Considerations:**

* **File Upload Safety**: Ensure that the uploaded files are scanned for any malicious content before processing.
* **Data Privacy**: The uploaded files are processed only temporarily on the server, and the server deletes them after the decryption process is complete.
* **Limitations**: This script is not guaranteed to work for all Office files, particularly if the password is set using advanced encryption techniques not supported by the script.

**Conclusion:**

This tool provides an efficient way to decrypt and remove password protection from VBA code embedded in Office files. By following the provided instructions, users can regain access to locked VBA projects, making the process of code review or modification much easier.