MY APPROACH TOWARDS BUILDING A SOFTWARE TOOL

1. **Backend:**

   Choose a backend framework (Django).

   Decide on the database (MongoDB) for storing website information and change history.

2. **Frontend:**

   Use a frontend framework (React) for the user interface.

3. **Implement Web Scraping:**

   Develop a module to fetch HTML content from websites.

4. **HTML Parsing:**

   Use HTML parsing libraries (BeautifulSoup) to extract relevant sections efficiently.

5. **Understand Cosmetic Filters:**
   - CSS Selectors:

     Allow users to define CSS selectors for the sections they want to monitor.

6. **Keyword Input:**

   Allow users to input keywords for each monitored section.

7. **Keyword Matching:**

   Develop a module to match content changes against specified keywords.

8. **Change Detection:**
   - Snapshot Comparison:

     Take periodic snapshots of the selected website sections.

     Compare current snapshots with previous ones to identify changes.

   - Algorithm for Change Detection:

     Develop an algorithm to determine if the changes are relevant based on the keywords.

9. **Alert System:**

   Implement a notification system for users when relevant changes occur.

10. **User Interface:**
    - Dashboard:

      Create a user-friendly dashboard for managing bookmarked websites, configuring monitoring settings, and viewing alerts.

    - Reports:

Include reporting features for historical changes.

11. Scalability:

Design the tool to handle a large number of websites and users.

12. Security:

Implement secure storage practices for user data and website content.

Use robust authentication mechanisms to protect user accounts.

13. Testing:

- Unit Testing:
  Test individual components for functionality and correctness.
- Integration Testing:
  Verify the interaction between different modules.
- User Acceptance Testing:
  Involve end-users to ensure the tool meets their expectations.

14. Documentation:

Provide comprehensive documentation for users.

Document the codebase for future maintenance.

15. Deployment:

Choose a hosting solution (AWS) for deployment.

Implement continuous integration and deployment for seamless updates.

16. Maintenance And Updates:

Feedback Loop:

Establish a feedback loop for users to report issues and suggest improvements.