Team Details

a. **Team name:** SpoofShield
b. **Team leader name:** Sarish Sonawane
c. **Domain :** Security
d. **Problem Statement:** Email Spoofing Prevention Checker

Proposed Solution

1. Dual Approach:
- Website: Enables users to upload email files for in-depth analysis.
- Browser Extension: Integrates with email platforms for real-time email spoofing detection.

2. Key Features:
- Automated SPF, DKIM, and DMARC validation to detect spoofing.
- Analysis of email headers for suspicious patterns and vulnerabilities.
- User-friendly interface for seamless email security checks.

3. Multi-Layered Security:
- Incorporates additional checks for **SPF, DKIM, and DMARC** misconfigurations, ensuring a robust defense against various spoofing techniques.

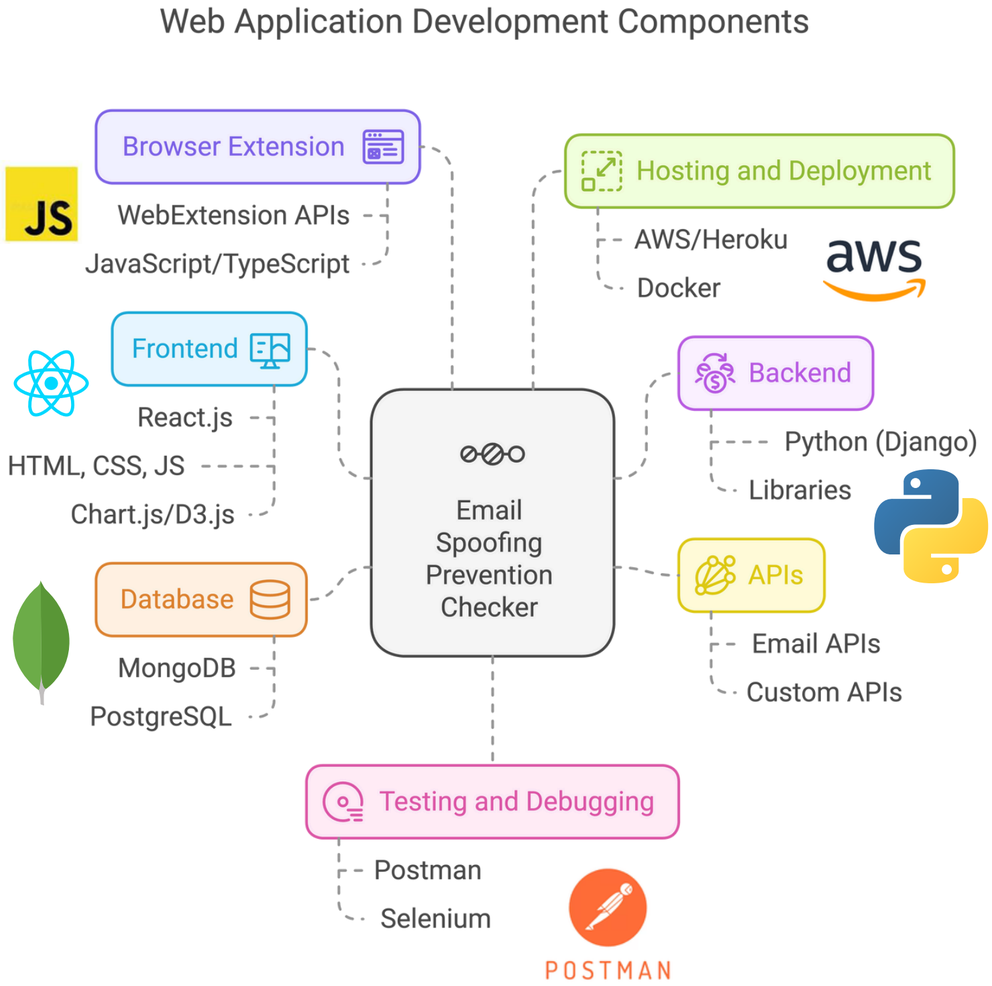
4. Actionable Insights:
- Provides users with recommendations for improving email security, such as adjusting DNS settings and configuring authentication protocols correctly.

Innovation & Uniqueness

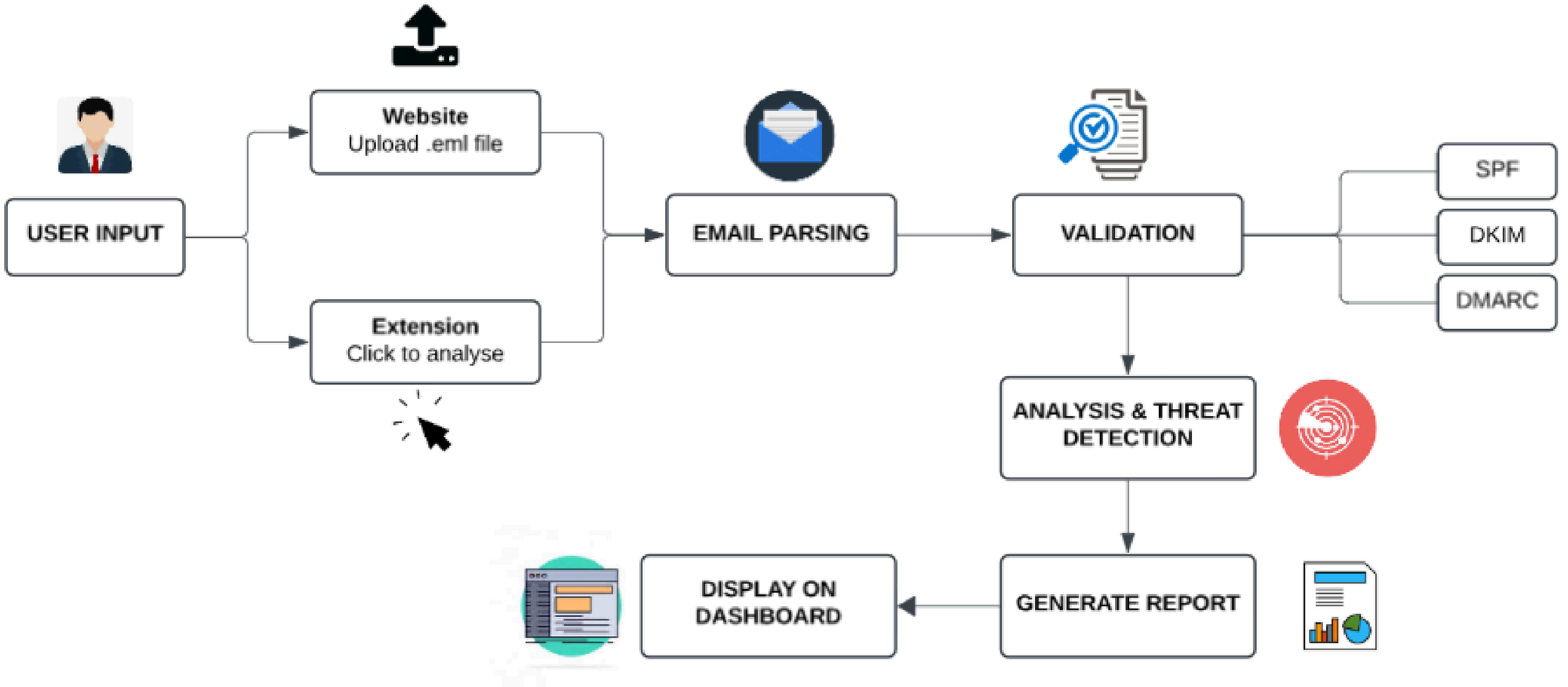
- **Website** + **Extension** Combination for flexibility and accessibility.
- **Automated header extraction** via browser extension for effortless analysis.
- **AI Driven insights**
- **Compatibility** with **multiple** email providers (e.g., **Gmail**, **Outlook**, **Yahoo**) for a seamless user experience.
- Detailed **feedback** and actionable **recommendations** for enhanced email security.

TECHNICAL APPROACH

1. **Frontend:**
   - React.js: Builds responsive user interfaces.
   - Chart.js/D3.js: Visualizes analysis results dynamically.
2. **Backend:**
   - Flask/Django: Handles logic, email parsing, and validation.
   - Libraries:
     - dnspython, dkimpy, pydmarc: For SPF, DKIM, and DMARC validation.
     - email.parser: Extracts and analyzes email headers.
3. **APIs:**
   - Gmail/Outlook APIs: Retrieves email data.
   - Custom APIs: Processes headers and returns validation results.
4. **Database**
   - MongoDB/PostgreSQL: Optionally stores email data and analysis history.
5. **Browser Extension**
   - WebExtension APIs: Chrome/Firefox integration.
   - JavaScript/TypeScript: Implements extension logic.
6. **Deployment**
   - AWS/Heroku: Hosts the website and backend.
   - Docker: Ensures containerization and smooth deployments.
7. **Testing**
   - Postman: For API functionality testing.
   - Selenium: Validates browser extension workflows.



FLOW DIAGRAM :-

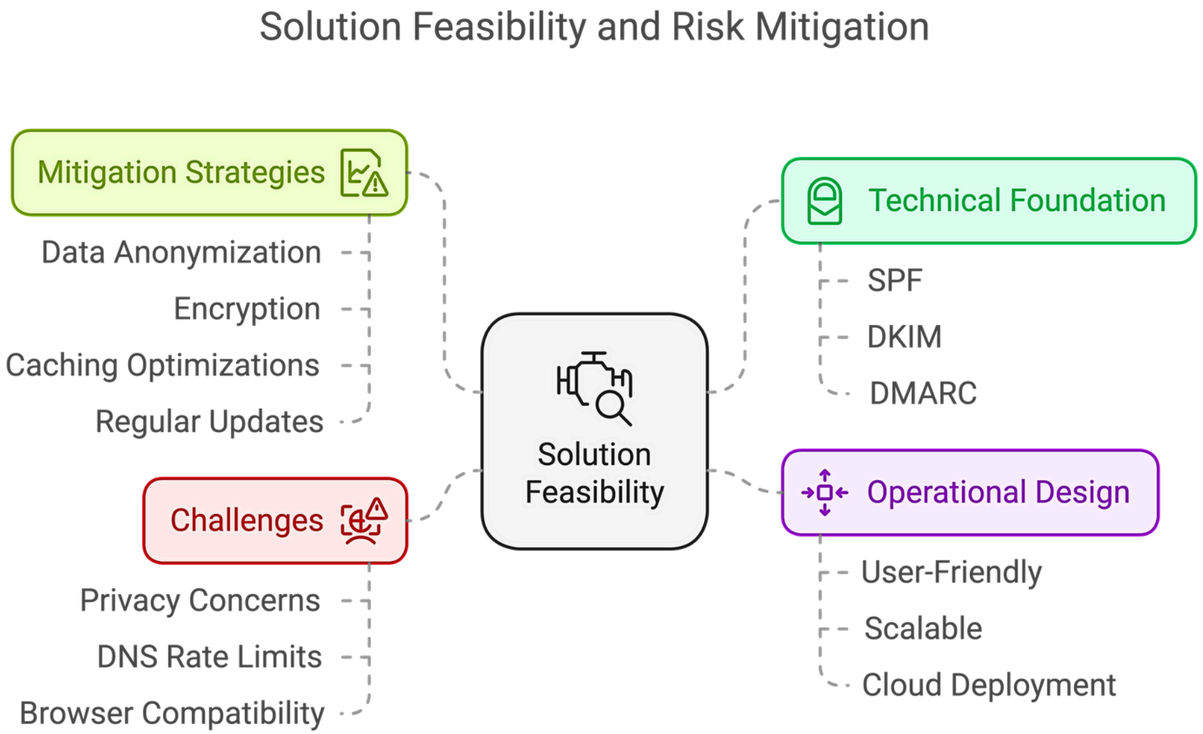


FEASIBILITY AND VIABILITY

1. Feasibility Analysis:
- Technical: Uses well-established protocols (SPF, DKIM, DMARC) and libraries (Python, WebExtension APIs).
- Operational: User-friendly with minimal input required.
- Financial: Scalable and cost-effective with cloud deployment (AWS/Heroku).

2. Potential Challenges and Risks:
- Privacy Concerns: User hesitation to upload sensitive emails.
- DNS Limits: Risk of rate limits with high query volume.
- Extension Compatibility: Browser-specific restrictions.
- Evolving Threats: Constantly changing spoofing techniques.

3. Strategies to Overcome Challenges:
- Privacy: Anonymize and encrypt user data.
- DNS Optimization: Cache results and optimize queries.
- Compatibility: Maintain extensions for multiple browsers.
- Updates: Regularly update detection algorithms.



Team Details

Sarish Sonawane : Second Year (Information Technology)
Email: sarishsonawane2005@gmail.com | Mobile: 9922258259

Atharva Dhavale : Second Year (Computer Engineering)
Email: atharva18dhavale@gmail.com | Mobile: 8329278975

Aayush Meghal : Second Year (ENTC)
Email: meghalaayush@gmail.com | Mobile : 9405417042

Anuj Nagpure : Second Year (Computer Engineering)
Email: anujnagpure6@gmail.com | Mobile: 7058630684