Google Summer of Code 2025
Proposal

Ccextractor

Project Title : URL Shortener with a Twist

Personal Information

Name: Utkarsh Chandrakant Kadu

Email: kaduutkarsh5@gmail.com

GitHub Profile: Utkarsh8867 (Utkarsh Kadu)

LinkedIn Profile: Utkarsh Kadu | LinkedIn

Location : India

University : Savitribai Phule Pune University(SPPU)

Degree Program: Bachelor of Engineering in Computer Engineering

Graduation Year: 2026

Background

I am currently pursuing a Bachelor of Engineering degree in Computer Engineering at Parvatibai Genba Moze College of Engineering, Pune. I have a strong passion for open-source development, particularly in areas of software engineering, data structures, and NLP.

I previously contributed to open-source projects, refining my skills in backend and API development. My experience with frameworks such as React.js, Next.js, and database management systems like Firebase and PostgreSQL enables me to design and develop scalable applications.

Skills

Languages: Python, Java, JavaScript, C++,Dart
Frameworks: React.js, Next.js, Node.js, Django
Databases: Firebase, PostgreSQL, MongoDB
Other Tools: Git, Docker, Cloudflare, Redis

Why Am I Interested in CCExtractor?

In the world of video accessibility, CCExtractor plays a crucial role in providing accurate subtitle extraction solutions for different media formats. Unlike many automated subtitle generation tools, CCExtractor focuses on precise extraction without relying entirely on machine learning, making it an invaluable tool for low-latency environments.

Its open-source nature, community-driven development, and potential to enhance accessibility across various platforms make it an exciting project to contribute to. By improving and optimizing CCExtractor, I aim to contribute to a more inclusive and user-friendly experience for all users.

Synopsis

This project aims to develop a URL shortener with unique features focused on privacy and access control. Unlike traditional URL shorteners that focus on mass accessibility, this tool will generate links intended for a small group of users while maintaining confidentiality about the URL's content.

Benefits to the Community

Many URL shorteners exist, but few focus on security, user-specific access, and privacy. This project will enable users to:

- Restrict access to specific individuals via email authentication.
- Set expiration dates for links.
- Get notified when specific users open the links.
- Maintain a simple, easy-to-use interface while ensuring privacy.

This tool will be especially useful for personal sharing, internal team communications, and confidential information dissemination.

Deliverables

Phase 1

- Implement a backend system for generating and managing shortened URLs.
- Develop an authentication system using email-based verification.
- Design and implement a frontend UI for URL creation and management.
- Enable setting expiration and start dates for URLs.

Phase 2

- Implement email notifications when a link is accessed.
- Improve scalability by integrating a cloud-based solution (Firebase, Cloudflare Workers, etc.).
- Ensure security and efficiency in handling user authentication and URL storage.
- Conduct thorough testing and refine UI/UX for better usability.

Final Deliverable

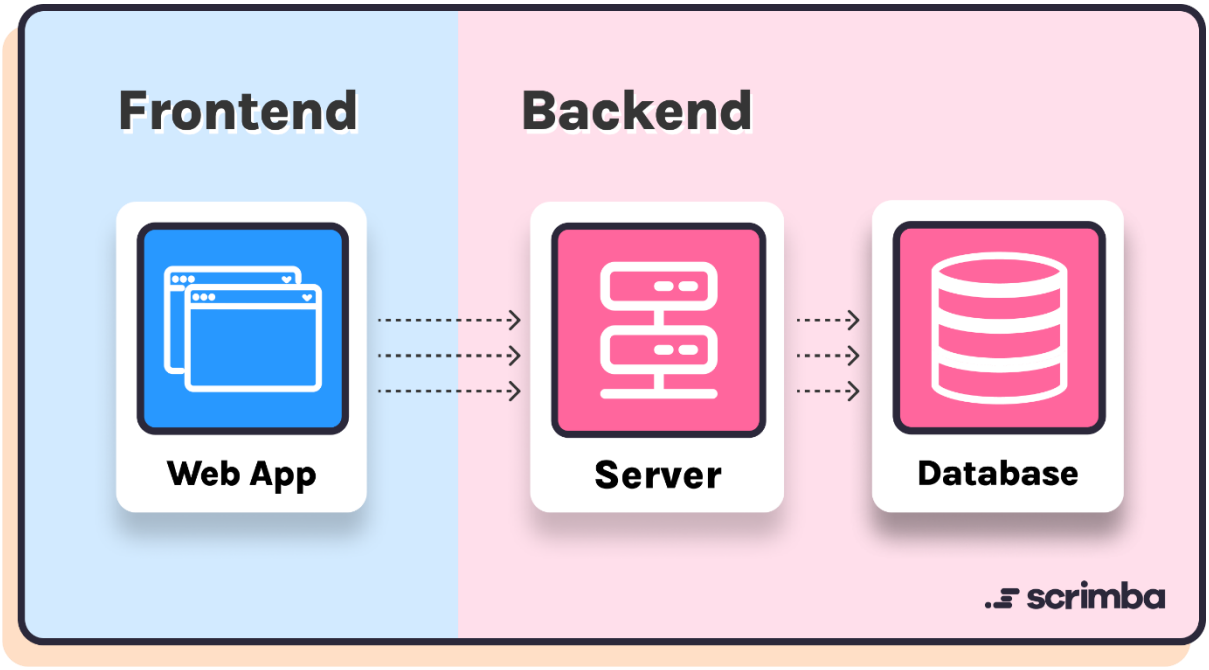
- A fully functional, open-source URL shortener with privacy-focused features.
- Documentation covering system architecture, API usage, and setup instructions.

Planning and Approach

System Architecture

The system will be divided into three major components:

1. **Frontend:** A React.js or Next.js web application for creating, managing, and tracking URLs.
2. **Backend:** A RESTful API built using Node.js (Express) or Django for URL processing, authentication, and storage.
3. **Database:** A cloud-based solution such as Firebase Firestore or PostgreSQL for storing URLs and user data.



Below is a high-level flowchart describing the process:

1. **User creates a short URL**
    - Inputs long URL and selects access control options.
    - Chooses expiration and start dates (optional).
    - Submits the form, generating a short URL.

2. **User shares the URL**
    - The system stores the short URL and its metadata.
    - If email authentication is required, an access list is stored.

3. **Recipient accesses the URL**
    - If authentication is required, the user receives an email confirmation.
    - Once verified, the original URL is revealed.

4. **URL tracking and notifications**
    - If tracking is enabled, the creator receives an email when the link is opened.
    - If expired, the system blocks access.

Technology Stack

- **Frontend:** React.js / Next.js for a dynamic UI
- **Backend:** Node.js (Express) or Django for server-side logic
- **Database:** Firebase Firestore / PostgreSQL for scalable data management
- **Authentication:** Magic link authentication via email (Firebase Auth / SendGrid)
- **Hosting & Deployment:** Vercel, Netlify, or Cloudflare Workers for reliable hosting

Scalability Considerations

- **Caching:** Use Redis or a similar caching mechanism for faster URL lookups.
- **Load Balancing:** Implement Cloudflare Workers for distributing traffic efficiently.
- **Security Measures:** Implement token-based authentication, rate limiting, and spam protection.

Timeline

Phase	Timeline	Milestone
Community Bonding	May 20 - June 17, 2025	Research best cloud solutions, finalize tech stack
Phase 1	June 17 - July 15, 2025	Implement backend, email auth, and basic UI
Phase 2	July 15 - August 12, 2025	Implement notifications, expiry system, scalability improvements
Final Submission	August 12 - August 19, 2025	Complete testing, documentation, and final refinements

---

Expected Outcomes

- A privacy-focused URL shortener with enhanced access control features.
- A simple UI/UX allowing easy URL management.
- Secure email-based authentication without complex logins.
- Scalable backend infrastructure to support real-world usage.

Why Me?

I have experience in full-stack web development, working with React.js, Next.js, and backend technologies like Node.js and Firebase. My background in database management and authentication systems makes me a strong candidate to implement this project successfully. I am also experienced in open-source contributions and enjoy building efficient and user-friendly applications.

Resources Required

- Access to Firebase, Cloudflare Workers, or alternative backend hosting.
- Email service provider for authentication (SendGrid, Firebase Auth, etc.).
- A testing environment to validate functionality across different devices.

Additional Information

- GitHub repository will be set up for code versioning and collaboration.
- The project will follow Agile development methodologies.
- Weekly updates will be provided to ensure transparency and continuous progress.

Links to Relevant Work

- Farmer's Market *(Demonstrates ability to build scalable web apps)*
- Create Next App *(Showcases frontend skills and form handling)*
- Utkarsh8867/DSA-Problems: DSA problem in Java. *(Demonstrates problem-solving and backend logic skills)*

Conclusion

This project is an opportunity to create a URL shortener with a unique value proposition. By focusing on user privacy, email-based authentication, and event notifications, we can develop a tool that stands out. My experience in full-stack development and passion for building impactful projects make me confident in delivering this successfully.

For any additional clarifications or discussions, feel free to reach out!

Thank you for considering my proposal! I look forward to contributing to this open-source initiative. 🚀