Patuakhali Science and Technology University
Faculty of Computer Science and Engineering

CIT 320 :: Software Development Project-II

Project Proposal

Project Title : Linkerine
Submission Date : 22 January 2026

Submitted from,	**Submitted to,**
Md. Sharafat Karim **ID** : 2102024, **Reg** : 10151, **Semester** : 5 (Level-3, Semester-1)	1. **Md Mahbubur Rahman** Associate Professor, Department of Computer Science and Information Technology, Patuakhali Science and Technology University. 2. **Md Atikqur Rahaman** Professor, Department of Computer Science and Information Technology, Patuakhali Science and Technology University.

Contents

Linkerine

1. Abstract

Linkerine is a modern, web-based bookmark and RSS management platform designed to help users seamlessly store, organize, and sync **bookmarks**, **articles**, **notes**, and **RSS feeds** across all devices. In an age of increasing digital information overload, Linkerine leverages the power of Next.js to provide a fast, SEO-friendly, and progressive web application (PWA) that offers a simplified yet powerful interface to manage and retrieve valuable resources from anywhere.

2. Objectives

- To build a high-performance, responsive web application for organizing bookmarks and RSS feeds.
- To implement secure server-side data fetching and storage using Next.js Server Actions.
- To provide an "Installable" experience on mobile and desktop via Progressive Web App (PWA) standards.
- To implement advanced features like automatic metadata scraping, nested collections, and full-text search.

3. Problem Statement

Existing solutions for bookmark management are often fragmented; users must choose between native apps that don't sync well with the web, or browser extensions that don't work on mobile.

Furthermore, many modern tools (like Pocket or Raindrop) lock essential features like full-text search or unlimited nested folders behind paywalls. There is a need for a unified, open-web solution that runs instantly on any browser while offering the performance of a native application.

4. Related Work

- **Mozilla Pocket** - Popular read-it-later app, but has become bloated with premium tiers. [1]
- **Goodlinks** - Excellent UI, but exclusive to the Apple ecosystem. [2]
- **Futurwise** - Chromium-only extension; lacks a standalone mobile interface. [3]

- **Instapaper** - Great for reading, but poor for organizing bookmarks. [4]
- **Raindrop.io** - The industry standard, but lacks self-hosting capabilities and anonymous usage. [5]

5. Scope

Linkerine will be developed as a modern web application using the **Next.js 15** framework. This ensures it is accessible from any device with a web browser (Windows, macOS, Linux, Android, iOS). The application will focus on providing a rich set of features including server-side RSS parsing, automatic Open Graph metadata extraction for links, and a responsive UI built with Tailwind CSS.

Later on, the project can be extended to include browser extensions that share the same codebase, public bookmark collections, and AI-driven content summarization.

5.1. Job Market Analysis

The job market for full-stack web developers in Bangladesh is growing rapidly, with significant demand for Next.js and modern JavaScript expertise. According to major job portals:

- **BD Tech Jobs** and **TechnTalents** consistently list 150-200+ open positions for full-stack developers, with a strong emphasis on Next.js, React, and TypeScript skills. [6], [7]
- **eJobs Bangladesh** and **JobMatchingBD** show that companies prioritize developers with experience in server-side rendering, database design, and PWA development—core features of Linkerine. [8], [9]
- Remote opportunities through **Arc.dev** and **Wellfound** indicate that specialized tools like Linkerine could attract both local and international users seeking productivity solutions tailored to the Asian market. [10], [11]
- **Careerjet Bangladesh** analysis reveals that bookmark management and RSS aggregation tools are in demand among knowledge workers and developers, suggesting a viable market niche for this project. [12]

The demand for full-stack developers with Next.js expertise is not just growing—it is becoming the standard for modern web development in Bangladesh, making Linkerine a timely project that demonstrates cutting-edge skill proficiency.

6. Methodology

6.1. Technology Stack

The development of Linkerine will follow an agile methodology. We have selected a stack optimized for performance, type safety, and rapid deployment.

Frontend Framework	Next.js 15 (React)
Language	TypeScript
Styling	Tailwind CSS + Shadcn/ui
Backend / API	Next.js Server Actions + Route Handlers

Database	PostgreSQL (via Supabase or Neon)
ORM	Drizzle ORM
Authentication	Clerk or Supabase Auth
Hosting	Vercel

Table 1: Technology Stack for Linkerine

6.2. Design Principles

The design of Linkerine will adhere to the following principles:

- **Mobile-First Responsive Design**: Ensuring the layout adapts perfectly from mobile screens to large desktop monitors using Tailwind CSS.
- **Server-Side Rendering (SSR)**: Utilizing Next.js to render pages on the server for speed and SEO, ensuring content is visible instantly.
- **Edge Computing**: Running middleware at the edge for low-latency authentication and redirect handling.
- **Type Safety**: Using TypeScript and Zod to ensure robust data validation from the database to the UI.
- **Accessibility**: Adhering to WCAG guidelines using the accessible primitives from Shadcn/ui (Radix UI).

7. Visual Models

7.1. ERD (Entity Relationship Diagram)

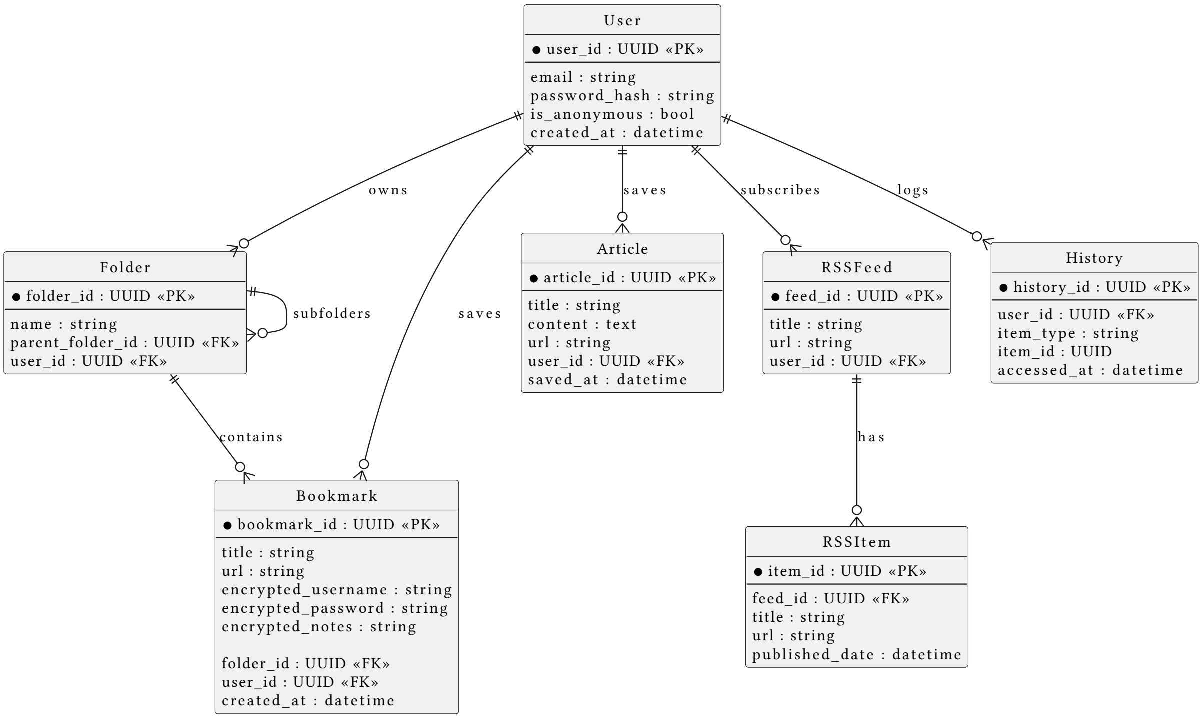


Figure 1: Entity Relationship Diagram of Linkerine

7.2. System Architecture Diagram

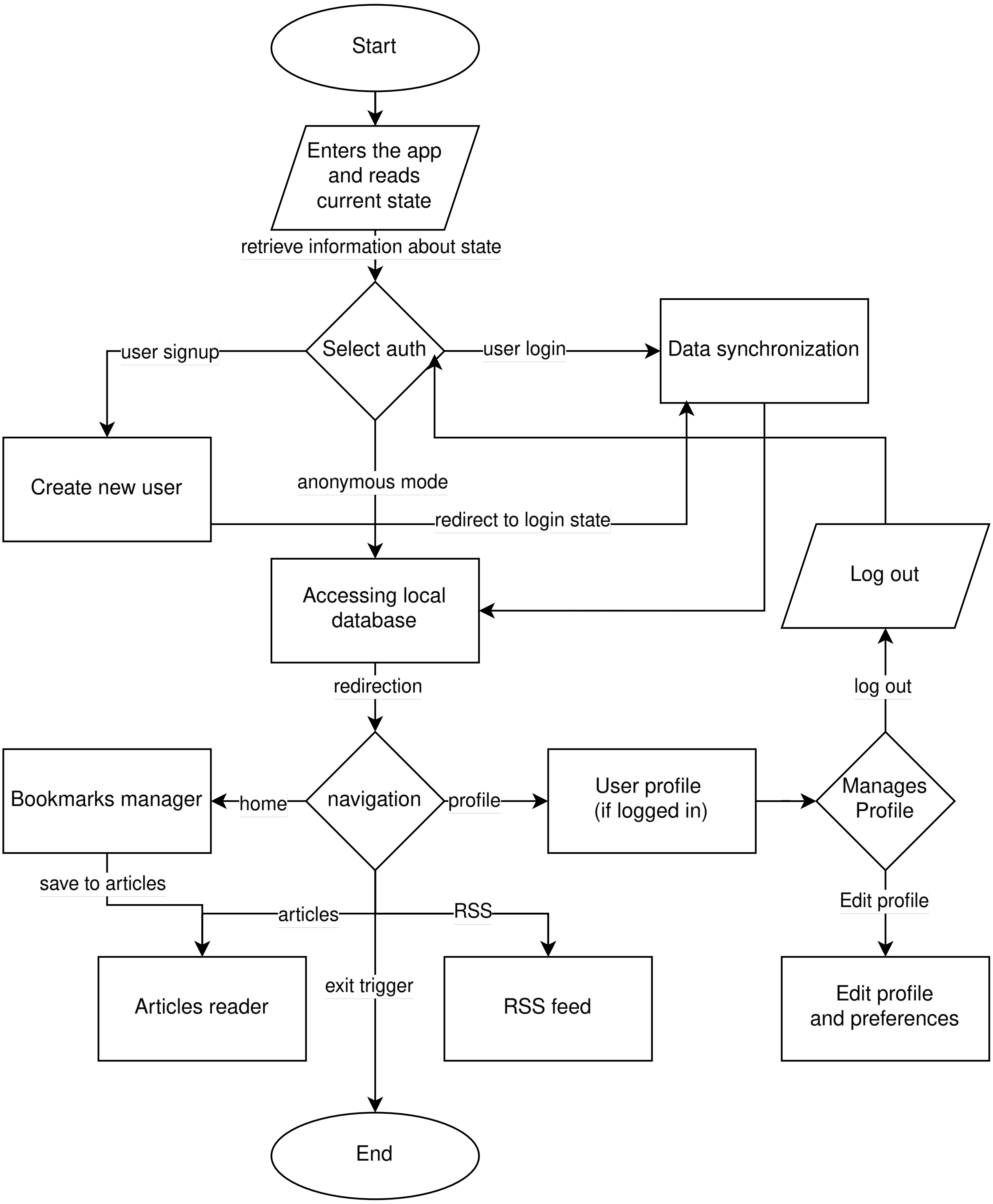


Figure 2: Next.js App Router Architecture & Data Flow

Here Figure 2 illustrates the architecture where the Next.js Client communicates with Server Actions, which then interact with the PostgreSQL database via Drizzle ORM. RSS feeds are fetched via a proxy route to handle CORS.

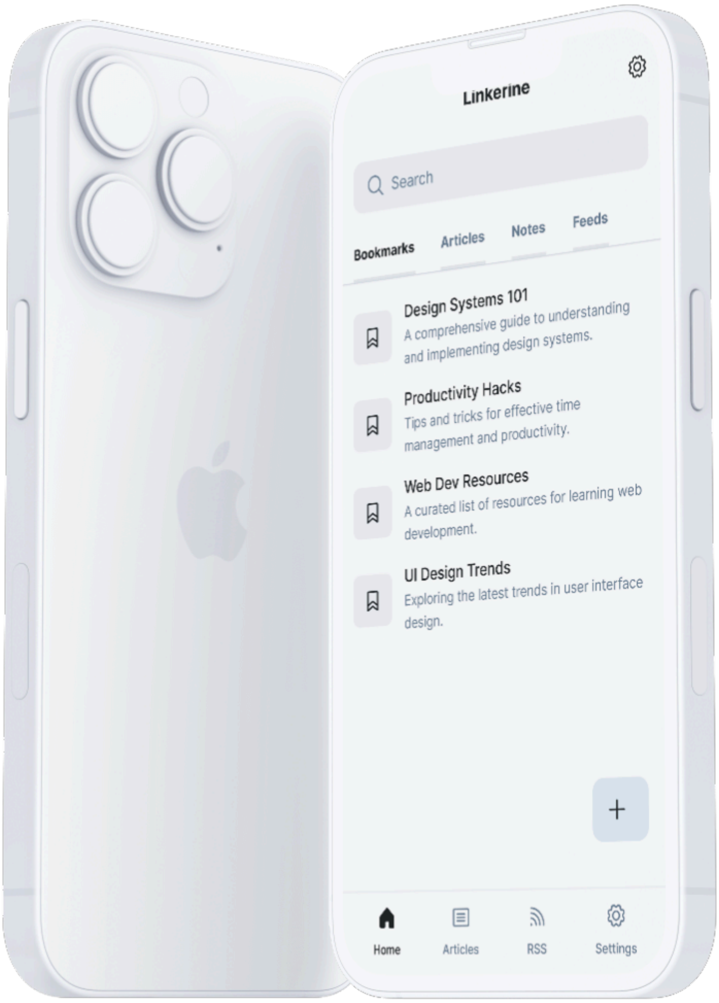
7.3. Timeline (Gantt Chart)

The timeline is divided into 12 weeks, focusing on the Next.js development lifecycle.

Task	Week 1-2	Week 3-4	Week 5-6	Week 7-8	Week 9	Week 10	Week 11	Week 12
Setup Next.js, TS, & Auth	✓		✓					
DB Schema (Drizzle) + Migrations		✓	✓					
Bookmark Actions & Scraping			✓	✓				
RSS Parser & Feed UI				✓				
Search & Tagging System			✓	✓	✓			
PWA Integration & Optimization						✓		
UI Polish (Dark Mode/Themes)						✓	✓	
Vercel Deployment & Testing								✓

Table 2: Development Timeline of Linkerine

8. UI Mockups



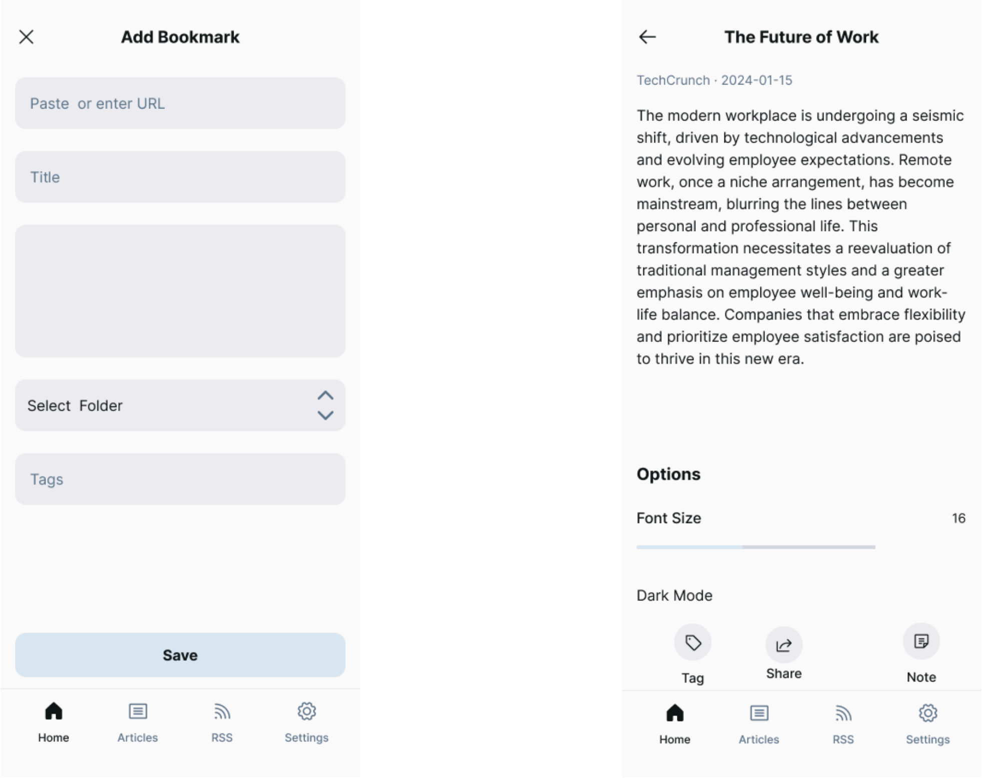
Design Systems 101
A comprehensive guide to understanding and implementing design systems.

Productivity Hacks
Tips and tricks for effective time management and productivity.

Web Dev Resources
A curated list of resources for learning web development.

UI Design Trends
Exploring the latest trends in user interface design.

Figure 3: Landing Page & Dashboard



The modern workplace is undergoing a seismic shift, driven by technological advancements and evolving employee expectations. Remote work, once a niche arrangement, has become mainstream, blurring the lines between personal and professional life. This transformation necessitates a reevaluation of traditional management styles and a greater emphasis on employee well-being and work-life balance. Companies that embrace flexibility and prioritize employee satisfaction are poised to thrive in this new era.

Options

Font Size: 16

Figure 4: Add Link Modal (with auto-fetch) and Reader View

9. Future Plans

1. **Browser Extensions**: Leveraging the shared TypeScript codebase to build Chrome/Firefox extensions that save links directly to the database.
2. **AI Summarization**: Integrating OpenAI or local LLMs to generate summaries of bookmarked articles.
3. **Public Collections**: Allowing users to publish specific folders as public, shareable web pages.
4. **Offline Sync**: Enhancing the PWA with local storage to allow reading cached articles without an internet connection.

10. Result

The expected outcome is a highly performant, server-rendered web application deployed on Vercel, providing a superior user experience for managing digital content across all device types.

11. References

[1] "Mozilla Pocket." Accessed: July 18, 2025. [Online]. Available: https://getpocket.com/

[2] "GoodLinks." Accessed: July 18, 2025. [Online]. Available: https://goodlinks.app/

[3] "Futurwise." Accessed: July 18, 2025. [Online]. Available: https://www.futurwise.com/

[4] "Instapaper." Accessed: July 18, 2025. [Online]. Available: https://instapaper.com/

[5] "Raindrop.io." Accessed: July 18, 2025. [Online]. Available: https://raindrop.io/

[6] "BD Tech Jobs: Specialized Software Engineering Portal." Accessed: Jan. 22, 2026. [Online]. Available: https://bdtechjobs.com/

[7] "TechnTalents: Automated ICT Job Platform Bangladesh." Accessed: Jan. 22, 2026. [Online]. Available: https://techntalents.com/

[8] "eJobs Bangladesh: Skill-Based IT Job Search." Accessed: Jan. 22, 2026. [Online]. Available: https://ejobs.com.bd/

[9] "JobMatchingBD: IT Careers and Internships in Dhaka." Accessed: Jan. 22, 2026. [Online]. Available: https://jobmatchingbd.com/

[10] "Arc.dev: Remote Senior Developer Roles in Bangladesh." Accessed: Jan. 22, 2026. [Online]. Available: https://arc.dev/

[11] "Wellfound: Startup and Full-Stack Roles in Bangladesh." Accessed: Jan. 22, 2026. [Online]. Available: https://wellfound.com/

[12] "Careerjet Bangladesh: Tech Job Search Engine." Accessed: Jan. 22, 2026. [Online]. Available: https://www.careerjet.com.bd/

THE END

12. Weekly Report

No.	Date	Objective	Status	Remarks
1		**Project proposal** • Initial research on existing solutions • Defining objectives and scope • Creating initial UI mockups		
2		**Environment and navigation setup** • Setting up Next.js development environment • Designing app structure with App Router • Creating basic UI components with Shadcn/ui		
3		**Basic UI and backend integration** • Implementing basic bookmark and RSS UI screens • Integrating Clerk/Supabase authentication • Setting up Tailwind CSS theming		
4		**Database schema and ORM setup** • Designing ERD with Drizzle ORM • Setting up PostgreSQL migrations • Implementing Server Actions for data operations		
5		**Bookmark and RSS feed features** • Implementing bookmark creation with metadata scraping • Building RSS parser and feed synchronization • Adding search and tagging system		
6		**PWA and UI Polish** • Integrating PWA capabilities • Implementing dark mode and responsive design • Optimizing performance for deployment		
7		**Final Testing & Deployment** • Conducting testing across devices and browsers • Fixing bugs and performance issues • Deploying to Vercel and finalization		