LinkVerse - Detailed Product Specification Document

1. Project Overview

Project Name: LinkVerse
Type: Mobile-first social bookmarking and micro-learning platform
Platform: Flutter (cross-platform: Android & iOS)
Purpose: Empower users to save, organize, and share useful learning links, while leveraging community-curated collections and AI-generated content summaries.

---

2. Goals & Objectives

- Provide users with a centralized platform to bookmark and organize educational resources.
- Enhance discovery of valuable learning materials via community contributions.
- Use AI to generate content summaries and categorize links intelligently.
- Allow micro-learning through bite-sized, curated content feeds.

---

3. Target Audience

- Students and self-learners
- Developers, designers, and researchers
- Educators and knowledge workers

---

4. Features & Requirements

4.1 User Authentication

- **Sign Up / Sign In:**
- Email/password
- Google OAuth2
- GitHub OAuth2
- **Password Recovery**
- **Onboarding:**
- Select topic interests
- Minimal walkthrough of core features

4.2 Bookmark Management

- **Add Link:**

- Input URL
- Auto-fetch metadata (title, description, image)
- User can add tags, custom title/notes
- **Organize Bookmarks:**
- Create folders/collections
- Move bookmarks across folders
- **Link View:**
- Web preview
- AI-generated summary (via GPT API)

4.3 Micro-Learning Feed

- **Daily Feed:**
- Personalized feed based on selected interests
- Option to "save for later", "mark as read"
- **Short Notes Mode:**
- View AI-generated summaries only

4.4 Community Collections

- **Create Public Collections:**
- Name, description, cover image
- Add bookmarks
- **Explore Collections:**
- View trending collections
- Follow, like, comment
- **Collection Moderation:**
- Admin can remove reported content

4.5 Search & Filtering

- **Search Bar:**
- Global search for bookmarks, collections, users
- **Filters:**
- By tags, domain, date added
- By popularity (likes/views)

4.6 User Profiles

- **Profile Page:**
- Username, bio, interests
- Public collections
- Followers/following

4.7 Analytics (Optional)

- **For Users:**
- Articles read per day/week
- Time spent reading

- **For Collections:**
- Views, likes, shares, saves

4.8 AI Integration

- **Summarization API (GPT/Gemini):**
- Article or video summary in ~3 sentences
- **Auto-tagging:**
- Suggest tags based on content

---

5. Technical Specifications

5.1 Frontend (Flutter)

- **Architecture:** MVVM with Riverpod
- **Navigation:** GoRouter
- **Storage:**
- Local: Hive or Drift
- Cloud Sync: Firebase Firestore or custom API
- **Packages:**
- `flutter_hooks`, `flutter_riverpod`, `http`, `url_launcher`, `firebase_auth`, `firebase_core`, `cached_network_image`

5.2 Backend Options

Option A: Firebase Stack

- Firebase Auth
- Cloud Firestore
- Firebase Functions (for AI calls, moderation)
- Firebase Storage (for user-uploaded images)

Option B: Custom Backend

- REST API using NestJS or Django
- PostgreSQL
- Supabase/Hasura for real-time sync
- Node.js AI microservice using OpenAI API

---

6. Architecture Diagram

Include a diagram showing:
- Frontend client (Flutter) communicating with Firebase or REST API - AI Microservice layer - Firestore/PostgreSQL as database - Optional Cloud Functions layer

---

7. Milestones & Timeline

Milestone	Description	Estimated Time
Requirements Gathering	Finalize feature list, use cases	3 days
UI/UX Design	Figma screens, user flow maps	5 days
Project Setup & Auth Integration	Firebase setup, auth screens	4 days
Bookmarking System	CRUD bookmarks, folder management	6 days
Community & Profiles	Public collections, follow system	7 days
Feed & Recommendation System	Personalized feed + AI summaries	5 days
AI Integration	Summary & tag suggestions	4 days
Search & Filters	Global search, filtering logic	3 days
Testing & QA	Unit/widget testing, bug fixing	5 days
Deployment	App store setup, Play Store & TestFlight builds	3 days

8. UX/UI Guidelines

- Use Material 3 Design principles
- Maintain color accessibility
- Responsive layouts for phones/tablets
- Use animations (e.g., Hero transitions) to enhance interactivity

9. Security Considerations

- Data encryption at rest & in transit
- Firebase security rules / backend authentication
- Input sanitization to prevent XSS/SQL injection (if using custom backend)
- Rate limiting on AI endpoints

10. Future Enhancements

- Chrome extension for quick saving
- Gamification: XP points, badges, reading streaks
- Group collections (e.g., classroom mode)
- Premium version with analytics, offline mode

11. License & Contributions

- License: MIT or Apache 2.0
- Contribution Guidelines: Open-source friendly (if public)

---

12. Deliverables

- Full source code (frontend + backend)
- API documentation (Swagger/Postman/Firebase Docs)
- Technical documentation & README
- Presentation/demo video
- Test report (unit, widget, and integration tests)

---

13. Team Structure (Example Roles)

- **Product Manager** – Requirements, priorities, roadmap
- **Flutter Developer(s)** – Frontend implementation
- **Backend Developer(s)** – API, DB, AI integration
- **UX/UI Designer** – Screens, flow, brand design
- **QA Tester** – Test planning and automation
- **DevOps (Optional)** – CI/CD, monitoring, deployment