

JejakAnak App - Comprehensive Masterplan V2.0

Executive Summary

JejakAnak adalah platform mobile yang membantu orangtua mendokumentasikan dan mengoptimalkan perjalanan tumbuh kembang anak dari masa kehamilan hingga remaja (prenatal - 15 tahun). Aplikasi ini menggabungkan curated educational content, activity logging, talent discovery melalui 4E framework (Enjoy, Easy, Excellent, Earn), gamification, dan community features untuk menciptakan ecosystem parenting yang engaging dan insightful.

Filosofi Nama: "Jejak Anak" melambangkan rekam jejak setiap langkah perjalanan anak, dimulai bahkan sebelum lahir, membentuk narasi lengkap tumbuh kembang yang dapat dilihat kembali di masa depan.

Value Proposition:

- **Untuk Orangtua:** Guidance terstruktur, dokumentasi memories, insights tentang minat/bakat anak
 - **Untuk Anak:** Pengalaman edukatif yang terpersonalisasi dan terdokumentasi
 - **Untuk Educator (Content Team):** Platform untuk mendistribusikan konten edukatif berkualitas
-

1. Vision & Objectives

1.1 Vision Statement

"Menjadi companion terpercaya orangtua Indonesia dalam membesarkan anak yang berkembang optimal sesuai potensinya, dengan mendokumentasikan setiap momen berharga dari kandungan hingga remaja."

1.2 Primary Objectives

1. **Documentation Excellence:** Menyediakan platform intuitif untuk mencatat milestone, aktivitas, dan memories anak dalam satu tempat
2. **Educational Guidance:** Menawarkan library aktivitas edukatif terstruktur, age-appropriate, dan ter-kurasi oleh ahli
3. **Talent Discovery:** Membantu orangtua mengidentifikasi minat dan bakat anak melalui 4E framework dan pattern recognition
4. **Parent Empowerment:** Memberikan insights actionable untuk mendukung perkembangan anak
5. **Community Building:** Memfasilitasi knowledge sharing antar orangtua (recipe exchange, play spots, tips)
6. **Long-term Engagement:** Menciptakan habit logging melalui gamification dan personalized recommendations

1.3 Success Metrics (12 Months Post-Launch)

User Acquisition:

- 50,000 registered users
- 25,000 monthly active users (MAU)
- 10,000+ children profiles created

Engagement:

- Average 3 activities logged per user per week
- 60% user retention rate (D30)
- 40% weekly active users (WAU/MAU ratio)

Content:

- 300+ curated activities across all age groups
- 20+ new activities added monthly
- 80% activity completion rate (users who start activity logging finish it)

Community:

- 500+ user-contributed content (recipes, tips, play spots)
- 5,000+ activity variations shared

Quality:

- Net Promoter Score (NPS) > 50
 - 4.5+ rating on Play Store & App Store
 - <2% churn rate monthly
-

2. Target Users

2.1 Primary Users: Parents & Guardians

Demographics:

- Age: 25-45 years old
- Income: Middle to upper-middle class (able to afford smartphones and activities)
- Location: Urban/suburban Indonesia (initially Jakarta, Surabaya, Bandung, Medan, Bali)
- Tech-savvy: Comfortable using mobile apps

Psychographics:

- **Engaged Parents:** Actively involved in child's development
- **Education-conscious:** Value early childhood education
- **Community-oriented:** Seek support from fellow parents
- **Documentation enthusiasts:** Love capturing moments (Instagram generation)

User Personas:

Persona 1: "Rina the First-Time Mom"

- Age: 28, pregnant with first child
- Works: Marketing professional
- Pain points: Overwhelmed with parenting information, unsure how to prepare
- Needs: Structured guidance, prenatal activities, community support
- Goals: Be a prepared parent, document pregnancy journey

Persona 2: "Budi the Active Dad"

- Age: 35, has 2 kids (4 and 7 years old)
- Works: Entrepreneur, flexible schedule
- Pain points: Runs out of activity ideas, wants quality time with kids
- Needs: Diverse activity suggestions, track progress
- Goals: Bond with kids, discover their talents early

Persona 3: "Siti the Organized Mom"

- Age: 32, has 1 child (5 years old)
- Works: Part-time consultant
- Pain points: Forgets to document moments, wants systematic tracking
- Needs: Easy logging, insights on child development
- Goals: Be intentional with child's education, preserve memories

2.2 Secondary Users: Educators & Content Creators

Uwa Farah's Content Team:

- Child development experts, psychologists, early childhood educators
- Role: Create and curate educational activities
- Needs: Easy-to-use content management tools with guidance
- Pain points: Not tech-savvy, need structured templates

Future Consideration: External educators, parenting coaches who might contribute content

2.3 Tertiary Users: Children (Indirect)

Age Range: Prenatal to 15 years

- **Prenatal (0 months):** Future child, activities for expectant mothers
- **Newborn (0-12 months):** Sensory stimulation, bonding activities
- **Toddler (1-3 years):** Motor skills, early exploration
- **Preschool (3-5 years):** Creativity, pre-literacy
- **Early Elementary (6-8 years):** Academic readiness, hobbies
- **Middle Childhood (9-11 years):** Skill mastery, interests
- **Pre-teen (12-15 years):** Identity, passion projects

Note: In V1, children do not directly operate the app. Parent-operated interface only.

3. Core Features (Version 1.0)

3.1 User Management & Profiles

User Registration & Authentication

- Email/password signup with email verification
- Social login: Google, Apple (OAuth 2.0)
- Password reset via email
- Secure token-based authentication (JWT)

Parent Profile

- Full name, profile photo
- Email, phone (optional)
- Location (city-level for content personalization)
- Notification preferences
- Privacy settings

Child Profile Management

- Multiple children per account (unlimited)
- Per child:
 - Name, nickname
 - Birth date OR Expected Due Date (EDD)
 - Status: "Expecting" or "Born"
 - Profile photo/avatar
 - Gender (optional)
 - Notes (special considerations, allergies, etc.)
- Auto-calculation of age in months
- Automatic content filtering by age
- Transition flow: When baby born, update status and unlock newborn content

3.2 Activity Library (Content Repository)

Content Structure

Each activity contains:

- **Basic Info:** Title, description (150 chars), slug
- **Classification:**
 - Age range (min/max in months)
 - Categories (multi-select: Art, Science, Outdoor, Physical, Music, Literacy, Cooking, Social)
 - Difficulty (Easy, Medium, Challenging)
 - Duration (15min, 30min, 1hr, 2hr+)
 - Location (Indoor, Outdoor, Both)
- **Materials Needed:** List with quantities, common vs special items flag
- **Instructions:** Numbered steps with optional images per step
- **Educational Value:**
 - Learning outcomes (Fine Motor, Gross Motor, Creativity, Problem Solving, Language, Social Skills, etc.)
 - Parent tips (how to facilitate)
 - Conversation starters (questions to ask child)
 - Safety notes (if applicable)
- **Media:** Main image, 0-5 additional images, optional video URL (YouTube/Vimeo)
- **Metadata:** Created by, status (draft/pending/published), publish date, popularity score

Browsing & Discovery

- **Home View:** Personalized recommendations (see section 3.3)
- **Library View:**
 - Grid or list display
 - Filters: Age range, category, difficulty, duration, location
 - Search: By title, keywords
 - Sort: Popular, newest, A-Z, duration
- **Category Pages:** Browse by specific category
- **Activity Detail:** Full-screen view with all content

Content Lifecycle

- Content created by admin team via Admin Panel (see 3.9)
- Status flow: Draft → Pending Review → Published
- Continuous updates: New activities added weekly (target: 5-10/week)
- Launch target: 100+ activities across all age groups

3.3 Personalized Recommendations

Recommendation Engine (Rule-Based for V1)

Algorithm Factors:

1. **Age-appropriate:** Filter by child's current age
2. **Category balance:** Suggest categories child hasn't tried recently
3. **Interest alignment:** Prioritize categories where child showed high "Enjoy" in 4E
4. **Skill progression:** Match difficulty to child's demonstrated capability
5. **Recency:** Avoid recently completed activities (30-day window)
6. **Popularity:** Slight boost for highly-rated activities
7. **Trending:** Highlight new content (published within 2 weeks)

Presentation:

- "Recommended for [Child Name]" section on home dashboard
- 5 activity cards with reasons: "Because Rina loves Art activities" or "Time to try something new!"
- Refresh daily or after each logged activity

Future Enhancement (V2): Machine learning model for deeper personalization

3.4 Activity Scheduling

Schedule Feature

- From activity detail page: "Schedule" button
- User selects:
 - Date (calendar picker)
 - Time (optional, time picker)
 - Reminder: 15min / 1 hour / 1 day before
 - Personal notes (e.g., "Prepare materials the night before")
- Scheduled activities appear in:
 - "Upcoming Activities" section on dashboard
 - Calendar view (optional feature)
- Edit or cancel schedule anytime

Reminders

- Push notification at scheduled time minus reminder offset
- Notification content:
 - "🕒 Reminder: 'Finger Painting' activity in 1 hour!"
 - Action: Tap to open activity detail
 - Option to reschedule or mark as done

3.5 Activity Logging & Journaling

Core Logging Flow

Step 1: Initiate Log

- From activity detail: "Mark as Done" button
- Or from scheduled activity: Quick log shortcut

Step 2: Upload Photos

- Camera integration: Take photo directly
- Gallery picker: Select 1-10 existing photos
- Per photo: Optional caption
- Auto-compression before upload (max 2MB per photo)
- Photos stored in cloud (S3) with CDN

Step 3: Write Notes

- Free-text field (max 500 characters)
- Placeholder prompts: "How did [child] enjoy this? Any highlights?"
- Optional: Voice-to-text for convenience

Step 4: 4E Framework Feedback Visual toggles for:

- 😊 **Enjoy:** Did they love it?
 - Options: Yes / Neutral / No
- ✨ **Easy:** Was it challenging?
 - Options: Too Easy / Just Right / Challenging
- 🌟 **Excellent:** Did they excel?
 - Options: Yes / Okay / Struggled
- 💎 **Earn:** Did they create/produce something?
 - Options: Yes / No

Step 5: Additional Options

- Mark as favorite (❤️)
- Tag location (if applicable, link to play spot)
- Add custom tags (optional)

Step 6: Save

- Confirmation animation with confetti
- XP points awarded (+10 base, +5 for photo, +5 for detailed notes)
- Check for badge unlocks
- Prompt: "Share your variation?" (if applicable)

Activity Timeline (Journal View)

- Chronological display of all logged activities for a child
- Month/year section headers
- Each entry card shows:
 - Activity title & category
 - Date completed
 - Thumbnail photo(s)
 - Short preview of notes
 - 4E badges (icons showing ratings)
- Tap card to view full log details
- Filter timeline: By category, date range, favorites
- Export timeline: PDF or shareable image (future feature)

3.6 Child Dashboard & Insights

Dashboard Layout

Header:

- Child photo, name, age
- Current streak: "🔥 5-day streak!"
- Level badge: "Level 3 - Active Parent"

Quick Stats Cards:

- Total activities completed
- Categories explored (e.g., "6 out of 8")
- Photos uploaded
- Favorite category (most logged)

Activity Breakdown Chart:

- Pie chart or bar chart
- Shows distribution across categories
- Interactive: Tap segment to filter timeline

Talent Indicators Section:

- AI-generated insights (rule-based V1):
 - " 🎨 Rina shows strong creative talent!"
 - "Based on 8 Art activities where she excelled and enjoyed"
 - " 🧪 Consider science classes for Budi"
 - "He finds Science activities easy and often excels"
- Confidence level indicator (Low/Medium/High)
- Action button: "Explore more [category] activities"

Areas to Explore:

- Categories with low or zero activity count
- " 💡 You haven't tried Music activities yet! Here are 3 to start:"
- Carousel of recommended activities

Recent Memories:

- Last 3-5 logged activities with photos
- Quick access to timeline

Insights Generation Logic

Triggers:

- Recalculate after every 5 new activities logged
- Or weekly batch processing

Pattern Detection:

IF child has ≥ 10 activities in category X
 AND Enjoy rate in category X $> 70\%$
 AND Excellent rate in category X $> 50\%$
 THEN generate insight: "[Child] shows talent in [category]"
 CONFIDENCE: High if ≥ 15 activities, Medium if 10-14

IF child has 0 activities in category Y
 AND child's age is appropriate for category Y
 THEN generate suggestion: "Try [category Y]"

IF child's difficulty progression shows $80\%+$ "Just Right" or "Challenging"
 THEN generate insight: "Ready for advanced activities in [categories]"

3.7 Gamification System

Experience Points (XP) & Levels

XP Sources:

- Log activity: +10 XP
- Add photo: +5 XP per photo (max +25)
- Write detailed notes (>50 chars): +5 XP
- Complete 4E feedback: +5 XP
- Try new category (first time): +20 XP
- Schedule activity in advance: +3 XP
- Share recipe/tip: +15 XP
- Help other parents (content marked helpful): +10 XP

Level System:

- Formula: $\text{Level} = \text{floor}(\sqrt{\text{XP} / 100}) + 1$
- Levels:
 - Lv 1 (0-100 XP): Newbie Parent 🌱
 - Lv 2 (100-400 XP): Engaged Parent 📅
 - Lv 3 (400-900 XP): Active Parent ⭐
 - Lv 4 (900-1600 XP): Super Parent 🏆
 - Lv 5 (1600+ XP): Legendary Parent 👑

Level-Up Benefits:

- Unlock special badge border color
- Notification: "👏 Level Up! You're now Level 3!"
- Social sharing: Share achievement on social media

Achievement Badges

Badge Types (20+ total):

- **Early Bird** 🌱 : Log first 5 activities
- **Memory Keeper** 📷 : Upload 20 photos
- **Storyteller** 📖 : Write 10 detailed notes
- **Art Explorer** 🎨 : Complete 10 Art activities
- **Science Guru** 🔬 : Complete 10 Science activities
- **Outdoor Adventurer** 🌲 : Complete 10 Outdoor activities
- **Well-Rounded** 🌐 : Try all 8 categories
- **Week Warrior** 🔥 : 7-day activity streak
- **Month Master** 📅 : 30-day activity streak
- **Century Club** 100 : Log 100 activities
- **Helpful Parent** 💡 : Share 5 tips/recipes
- **Community Star** ★ : 20 contributions marked helpful
- **Early Adopter** 🚀 : Join during launch month
- **Consistency Champion** ⚡ : Log activities 4 weeks consecutively
- **Photo Pro** 📷 : Upload 100 photos
- **Category King/Queen**: Master one category (50 activities)
- **Talent Scout** 🎯 : Unlock 3 talent insights for child

Badge Display:

- Profile showcase page (grid of earned badges)
- Grayscale locked badges (show progress toward unlock)
- Badge detail modal: Description, earn date, rarity %
- Social sharing: "I just earned [badge]!" template

Streaks

Activity Streak:

- Counts consecutive days with at least 1 logged activity
- Display on dashboard: " 🔥 7-day streak! Keep it going!"
- Notification if streak at risk: "Don't break your 10-day streak! Log an activity today."
- Streak freeze: 2 free freeze days per month (auto-applied on first miss)

Milestone Celebrations:

- 7 days: Badge + encouraging message
- 30 days: Badge + feature in app spotlight
- 100 days: Legendary badge + special reward (unlock premium content?)

Monthly Challenges

Challenge System:

- Admin creates challenges each month
- Examples:
 - "October Outdoor Adventure": Complete 5 outdoor activities
 - "November Creativity Month": Try 3 different Art activities
 - "December Family Time": Log 10 activities total
- User joins challenge (opt-in)
- Progress tracker: "3/5 completed"
- Rewards: Exclusive badge, bonus XP, feature in challenge leaderboard

Challenge Display:

- Dedicated "Challenges" tab in app
- Active challenges at top
- Past challenges (with completion status)
- No competitive leaderboards (to avoid pressure)

3.8 Notification System

Notification Types

1. Scheduled Reminders

- Trigger: X minutes before scheduled activity
- Content: "🕒 [Activity name] starts in [time]! Ready to begin?"
- Actions: [Open Activity] [Snooze] [Cancel]

2. Engagement Nudges

- Trigger: 3 days without activity logging
- Content: "Miss you! ✨ [Child name] would love trying this: [Activity suggestion]"
- Frequency: Max once per week

3. Weekly Digest

- Trigger: Every Monday 9 AM
- Content: "Week ahead! 5 new activities added + Personalized picks for [Child]"
- Includes: Activity cards, child's stats summary

4. Milestone Celebrations

- Trigger: Achievement unlocked
- Content: "🎉 Wow! [Child] completed 10 activities! You're amazing."
- Actions: [View Progress] [Share]

5. Badge Earned

- Trigger: Immediate on badge unlock
- Content: "Achievement Unlocked! 🏆 [Badge name]"
- Actions: [View Badge] [Share]

6. New Content Alerts

- Trigger: Weekly content publish
- Content: "[X] new activities perfect for [Child age]! 🎯"
- Frequency: Weekly (Fridays)

7. Streak Warnings

- Trigger: 18 hours since last activity if streak active
- Content: "🔥 Keep your [X]-day streak alive! Quick activity idea: ..."
- Frequency: Max once daily

8. Community Updates

- Trigger: User's contribution approved
- Content: "Your recipe is live! 🍪 Other parents will love it."
- OR: "5 parents found your tip helpful! +50 XP"

9. Insight Generation

- Trigger: New insight computed
- Content: "💡 New insight about [Child]: [Insight summary]"
- Actions: [View Full Insight]

10. Challenge Updates

- Trigger: Progress milestones
- Content: "Halfway there! 3 more activities to complete October Challenge."

User Controls

- Notification preferences in settings:
 - Toggle each type on/off
 - Quiet hours (e.g., 10 PM - 7 AM)
 - Delivery channel: Push / Email / Both
- Default: All enabled except engagement nudges (opt-in)

Implementation

- **Push Notifications:** Firebase Cloud Messaging (FCM)
- **Email Notifications:** SendGrid or AWS SES (for digests, important alerts)
- **In-App Notifications:** Bell icon with badge count, dropdown list

3.9 Admin Panel (Content Management)

Access & Roles

Admin Roles:

- **Super Admin:** Full access (tech team)
- **Content Manager:** Create, edit, publish content (Uwa Farah)
- **Content Creator:** Create, submit for review (Uwa Farah's team)
- **Reviewer:** Approve/reject user-generated content

Authentication:

- Separate login from parent app
- Role-based access control (RBAC)
- Activity logging for audit trail

Features

Dashboard Home:

- Quick stats:
 - Total activities published
 - Pending review count
 - This week's user completions
 - Most popular activity (trending)
- Recent user activity feed (sample logs)
- Quick actions: Create Activity, Review Queue

Create/Edit Activity:

Step 1: Choose Template

- Pre-structured templates by category:
 - 🎨 Art & Craft Activity
 - 🧪 Science Experiment
 - 🏃 Physical Activity
 - 📖 Literacy & Reading
 - 🍴 Cooking & Nutrition
 - 🎵 Music & Rhythm
 - 🌳 Outdoor Exploration
 - ✨ General/Custom
- Each template has pre-filled sections with placeholders

Step 2-5: Form Wizard (as detailed in discussion)

- Basic Info (title, age, categories, difficulty, duration, location)
- Materials & Prep
- Instructions (steps with image uploads)
- Educational Value (learning outcomes, tips, conversation starters)
- Media & Preview

Validation System:

- Required field checks
- Character limits enforced
- Image size/format validation
- Warning for missing recommended fields
- Preview mode: See mobile app view before publish

Bulk Import:

- CSV/Excel template download
- Upload file with parsed data
- Preview import with error highlights
- Bulk create as drafts
- Review individually before publishing

Activity Library Management:

- List view with filters (status, category, date)
- Search by title
- Bulk actions: Publish, archive, delete
- Edit existing activities
- Duplicate activity (as starting template)
- View analytics per activity (completion rate, ratings)

Review Queue:

- List of pending content:
 - User-submitted recipes
 - Activity variations
 - Play spots
 - Parent tips
- Per item:
 - Content preview
 - Submitter info
 - Actions: Approve, Reject, Request Changes
 - Text field for reviewer notes
- Filter by content type, date, priority

Content Analytics:

- Activity performance:
 - Views, completion rate, average rating
 - 4E feedback aggregates
 - Most logged activities (trending)
- User engagement:
 - Total active users
 - Activities logged per day/week
 - Category popularity distribution
- Community contributions:
 - Submissions per week
 - Approval rate
 - Top contributors

User Management (Super Admin only):

- View registered users
- Search by email, name
- User activity summary
- Moderate reported content
- Ban/unban users (safety)

Challenge Management:

- Create monthly challenges
- Set criteria (type, target, duration)
- Define rewards (badge, XP)
- View participation stats
- End challenge and distribute rewards

Technical Implementation

- **Platform:** Web-based (responsive design)
- **Framework:** React or Vue.js SPA
- **Backend:** Same NestJS API (admin routes)
- **Authentication:** Separate JWT for admin users
- **Hosting:** Same infrastructure, different subdomain (admin.jejakanak.com)

Alternative Option: Leverage Strapi CMS

- Headless CMS with admin panel out-of-the-box
- Customize content types for Activity, etc.
- Faster setup, trade-off on customization
- Decision: Evaluate during Phase 0 (prototype)

3.10 Community Features (Phased Rollout)

Phase 1 (V1.0): Activity Variations

User Flow:

- After logging activity, optional prompt: "Try something unique? Share your variation!"
- Submit:
 - Short note (max 280 chars)
 - 1 photo (optional)
- Moderation:
 - Auto-approve if passes profanity filter
 - Manual review queue for edge cases
- Display:
 - On activity detail page, section "Other Parents' Ideas"
 - Carousel of variation cards
 - "Helpful" button (upvote)

Safety:

- No child face recognition enforcement (privacy)
- Report button for inappropriate content
- Admin review for flagged content

Phase 2 (V1.5): Recipe Exchange

User Flow:

- For cooking-category activities only
- Submit full recipe:
 - Title, ingredients (structured list), steps, photo
 - Prep time, servings, dietary tags (optional)
- Moderation:
 - Uwa Farah team reviews (food safety check)
 - Approve/reject with feedback
- Display:
 - "Parent Recipes" tab on cooking activities
 - Filter by dietary preference (future)

Incentive:

- Badge: "Recipe Contributor" after 1st approval
- Badge: "Master Chef" after 10 approvals
- XP rewards: +15 per approved recipe

Phase 3 (V1.5): Play Spots Database

User Flow:

- When logging outdoor activity, option to "Tag Location"
- Search existing spots OR add new:
 - Name, type (playground, park, mall play area, etc.)
 - Address (Google Maps autocomplete)
 - Facilities checkboxes (parking, restroom, food, wheelchair access)
 - Age suitability estimate
 - Photo (optional)
 - Short note (max 200 chars)
- Auto-publish (lightweight moderation)

Discovery:

- "Play Spots Near You" section in app
- Map view with pins
- List view with distance, rating (upvote count)
- Filter: Type, age, facilities
- Each spot shows:
 - Activities logged here (count)
 - Parent notes/reviews
 - [Get Directions] button (Google Maps deep link)

Community Curation:

- Upvote/downvote system
- Admin can merge duplicates or remove spam

Phase 4 (V2.0+): Parent Tips Forum

Structure:

- Categories: Activity Hacks, Nutrition, Sleep, Books, General
- User posts: Title, body (max 1000 chars), optional photo
- Tags (pre-defined)
- Engagement: Helpful button, Save/bookmark, Comments

Moderation:

- Pre-moderation (all posts reviewed before publish)
- Uwa Farah team can feature best posts
- Clear community guidelines

Future: Direct messaging, parent groups (with heavy safety considerations)

4. Technology Stack

4.1 Mobile Application

Framework: Flutter (Dart)

Rationale:

- Cross-platform (Android & iOS from single codebase)
- Near-native performance
- Hot reload for rapid development
- Mature ecosystem (50,000+ packages)
- Large developer community (1M+ active developers globally)
- Strong documentation and Google backing
- AI coding assistants (Claude, Codex) well-trained on Dart/Flutter

Key Libraries:

- **State Management:** Riverpod or Provider (proven, simple)
- **Networking:** Dio (HTTP client with interceptors)
- **Local Storage:** Hive or SharedPreferences (lightweight)
- **Image Handling:** image_picker, cached_network_image, flutter_image_compress
- **Navigation:** go_router (declarative routing)
- **UI Components:** Material Design 3 widgets
- **Charts:** fl_chart (for dashboard visualizations)
- **Maps:** google_maps_flutter (for play spots)
- **Notifications:** firebase_messaging (FCM)
- **Camera:** camera package
- **Analytics:** firebase_analytics
- **Crash Reporting:** firebase_crashlytics

Architecture Pattern: Clean Architecture + Feature-First Structure

```

lib/
├── core/
│   ├── network/
│   ├── storage/
│   ├── utils/
│   └── theme/
├── features/
│   ├── auth/
│   │   ├── data/ (repositories, models)
│   │   ├── domain/ (entities, use cases)
│   │   └── presentation/ (screens, widgets, providers)
│   ├── activities/
│   ├── logging/
│   ├── dashboard/
│   └── ...
└── main.dart
  
```

4.2 Backend Server

Runtime: Node.js (v20 LTS)

Framework: NestJS (TypeScript)

Rationale:

- Modular architecture by design
- TypeScript for type safety and maintainability
- Built on Express (proven web server)
- Dependency injection (enterprise-grade)
- Extensive middleware ecosystem
- Well-documented, widely adopted
- AI coding assistants proficient in TypeScript

Project Structure:

```
src/
├── modules/
│   ├── auth/
│   ├── users/
│   ├── children/
│   ├── activities/
│   ├── activity-logs/
│   ├── recommendations/
│   ├── gamification/
│   ├── notifications/
│   ├── community/
│   └── admin/
├── common/
│   ├── decorators/
│   ├── filters/
│   ├── guards/
│   ├── interceptors/
│   └── pipes/
├── config/
├── database/
└── main.ts
```

Key Libraries:

- **ORM:** Prisma (modern, type-safe database access)
- **Validation:** class-validator, class-transformer
- **Authentication:** @nestjs/jwt, @nestjs/passport
- **File Upload:** multer, AWS SDK (for S3)
- **Task Scheduling:** @nestjs/schedule (for cron jobs)
- **Caching:** @nestjs/cache-manager (with Redis)
- **Logging:** winston, morgan
- **Testing:** Jest (unit & integration tests)
- **API Documentation:** @nestjs/swagger (auto-generated docs)
- **Email:** nodemailer or SendGrid SDK

API Design:

- RESTful principles
- Versioned endpoints: `/api/v1/...`
- Consistent response format:

```
json
{
  "success": true,
  "data": { ... },
  "message": "...",
  "timestamp": "2025-10-20T..."
}
```

- Error handling: Standard HTTP codes + descriptive messages
- Rate limiting: By IP and user (prevent abuse)
- CORS enabled for mobile app

4.3 Database

Primary Database: PostgreSQL 15+

Rationale:

- ACID compliance (data integrity critical for children's data)
- Robust relational model fits our data structure
- JSONB support (flexibility for 4E feedback, nested data)
- Most popular database among developers (49% usage)
- Excellent performance and scalability
- Rich ecosystem of tools and extensions
- Proven in production at scale

Schema Management:

- Prisma ORM for schema definition and migrations
- Version-controlled migrations
- Seed scripts for initial data (categories, badges, etc.)

Key Tables (see Section 6 for full schema):

- users, children, activities, activity_logs, scheduled_activities
- user_badges, challenges, child_insights
- play_spots, activity_variations, parent_recipes
- notifications, push_tokens

Database Hosting:

- **Production:** AWS RDS PostgreSQL (managed service)
 - Automated backups (daily)
 - Multi-AZ deployment (high availability)
 - Read replicas (if needed for scaling)
- **Development:** Local PostgreSQL or Docker container

Caching Layer: Redis (optional for V1, recommended for V2)

- Cache frequently accessed data (categories, featured activities)
- Session storage (if needed)
- Rate limiting counters
- Leaderboard/stats (if real-time needed)

4.4 File Storage & CDN

Storage: AWS S3

- Photo uploads stored in private buckets
- Organized by user/child ID
- Lifecycle policies: Move old photos to cheaper storage (Glacier)

CDN: AWS CloudFront

- Fast global delivery of images
- Signed URLs for private content (child photos)
- Caching strategy: Long TTL for activity images, shorter for user content

Image Processing:

- Client-side compression (Flutter)
- Optional: AWS Lambda for additional resizing/optimization on upload

4.5 Cloud Infrastructure

Provider: AWS (Amazon Web Services)

Rationale:

- Market leader, most mature cloud platform
- Comprehensive service catalog
- Strong documentation and community
- Compliance certifications (important for children's data)
- Regional availability (ap-southeast-1 for Indonesia)

Services Used:

- **Compute:** ECS Fargate (containerized backend, serverless)
- **Database:** RDS PostgreSQL
- **Storage:** S3 + CloudFront CDN
- **Networking:** VPC, ALB (Application Load Balancer)
- **Monitoring:** CloudWatch (logs, metrics, alarms)
- **Secrets:** AWS Secrets Manager (API keys, DB passwords)
- **Email:** SES (transactional emails)

Alternative Consideration: Google Cloud Platform or Azure (if team has existing expertise)

4.6 Push Notifications

Service: Firebase Cloud Messaging (FCM)

Rationale:

- Free
- Cross-platform (Android, iOS, Web)
- Reliable delivery
- Topic-based and targeted messaging
- Well-documented Flutter integration

Implementation:

- Backend sends messages via FCM Admin SDK
- Mobile app receives with `firebase_messaging` package
- Token management: Store device tokens in database
- Topics for broadcast: "all_users", "new_content", etc.

4.7 Analytics & Monitoring

User Analytics: Firebase Analytics + Google Analytics 4

- Event tracking (`activity_logged`, `badge_earned`, etc.)
- User demographics
- Retention cohorts
- Funnel analysis

Error Tracking: Sentry

- Real-time error alerts
- Stack traces with context
- Release tracking
- Performance monitoring

Application Monitoring: AWS CloudWatch + Grafana

- Server metrics (CPU, memory, request rate)
- Custom dashboards
- Alerting (Slack/email on critical events)

Logging: Winston (structured JSON logs) → CloudWatch Logs

- Centralized logging
- Searchable and filterable
- Retention policy: 90 days

4.8 CI/CD & DevOps

Version Control: Git + GitHub

CI/CD: GitHub Actions

- **Pipeline:**
 - On push to `dev` branch: Run tests, lint, build → Deploy to staging
 - On push to `main` branch: Run tests, build → Deploy to production
 - Manual approval for production deployments

Containerization: Docker

- Dockerfile for backend
- Docker Compose for local development (backend + PostgreSQL + Redis)

Infrastructure as Code: Terraform or AWS CloudFormation

- Define all AWS resources in code
- Reproducible environments (staging, production)
- Version-controlled infrastructure

Environments:

- **Development:** Local machines
- **Staging:** AWS environment (mirror of production, smaller scale)
- **Production:** AWS environment (auto-scaling, multi-AZ)

Deployment Strategy:

- Backend: Blue-green deployment via ECS (zero downtime)
- Mobile App: Phased rollout on Play Store/App Store (10% → 50% → 100%)

4.9 Third-Party Integrations

Maps: Google Maps API

- Geocoding for play spot addresses
- Map display in Flutter app
- Directions deep link

Social Login: Google Sign-In, Sign in with Apple

- OAuth 2.0 flow
- Reduced friction for user registration

Email Service: SendGrid or AWS SES

- Transactional emails (verification, password reset)
- Digest emails (weekly summary)

Payment Gateway (Future - if monetization):

- Midtrans (Indonesia-focused) or Stripe

4.10 Security Considerations

Data Protection:

- HTTPS everywhere (TLS 1.3)
- Encrypted database connections
- Encrypted S3 buckets (AES-256)
- Environment variables for secrets (never in code)

Authentication & Authorization:

- JWT with short expiry (1 hour) + refresh tokens
- Role-based access control (RBAC) for admin panel
- Password hashing: bcrypt (10 rounds)

Input Validation:

- All user input validated and sanitized
- Protection against: SQL injection, XSS, CSRF
- Rate limiting on API endpoints

Privacy:

- GDPR-compliant data handling (if targeting EU users)
- Privacy policy clearly displayed
- User data export feature (future)
- Account deletion cascade (remove all child data)
- Photos: Private by default, parent controls sharing

Child Safety:

- No public child profiles
 - Moderation for all user-generated content with children
 - Reporting mechanism for inappropriate content
 - No direct messaging in V1 (to prevent misuse)
-

5. Development Roadmap

5.1 Phase 0: Foundation & Validation (Week 1-3)

Week 1:

- [☒] Finalize requirements (this masterplan)
- [☒] Stakeholder alignment (team meeting)
- ☐ Create detailed user stories (product backlog)
- ☐ Design sprint kickoff

Week 2-3:

- ☐ Figma design: All key screens (20-25 screens)
- ☐ Build dummy web prototype (interactive, realistic data)
- ☐ Uwa Farah team: Prepare 30 seed activities (across age groups)
- ☐ Validation sessions:
 - Psikolog team review (2 hrs)
 - Educator team review (2 hrs)
 - Parent user testing (5 parents, 1 hr each)
- ☐ Iterate prototype based on feedback
- ☐ Final design sign-off

Deliverables:

- Figma design system + mockups
- Clickable web prototype
- Validation report (feedback summary)
- Prioritized feature backlog

5.2 Phase 1: MVP Alpha (Week 4-9) - 6 weeks

Goal: Core value loop functional

Sprint 1 (Week 4-5): Foundation

- ☐ Project setup (GitHub repos, CI/CD)
- ☐ Backend scaffolding (NestJS modules)
- ☐ Flutter app scaffolding
- ☐ Database schema + migrations
- ☐ Auth system (registration, login, JWT)
- ☐ User profile management
- ☐ Child profile CRUD

Sprint 2 (Week 6-7): Content & Browsing

- ☐ Activity model + API endpoints
- ☐ Admin panel V0.1 (basic activity creation)
- ☐ Seed 50+ activities (Uwa Farah team)
- ☐ Activity library UI (browse, filter, search)
- ☐ Activity detail page
- ☐ AWS S3 integration (image uploads)

Sprint 3 (Week 8-9): Logging & Timeline

- ☐ Activity logging flow (photos, notes, 4E)
- ☐ Timeline view (child's activity history)
- ☐ Activity scheduling feature
- ☐ Basic dashboard (stats cards)
- ☐ Push notification setup (FCM)

Testing:

- ☐ Unit tests for critical backend logic
- ☐ Integration tests (API endpoints)
- ☐ Flutter widget tests (key screens)
- ☐ Manual QA (test all flows)

Alpha Release:

- ☐ Deploy to staging environment
- ☐ Internal team testing (10 users)
- ☐ Bug bash session
- ☐ Iterate based on feedback

Deliverables:

- Functional MVP: Auth, Browse, Schedule, Log, Timeline
- 50+ activities in database
- Deployed on staging

5.3 Phase 2: MVP Beta (Week 10-13) - 4 weeks

Goal: Add intelligence & engagement

Sprint 4 (Week 10-11): Recommendations & Insights

- ☐ Recommendation algorithm implementation
- ☐ Insights generation logic (pattern detection)
- ☐ Enhanced dashboard (insights section)
- ☐ "Recommended for You" section on home

Sprint 5 (Week 12-13): Gamification & Admin

- ☐ Gamification system (XP, levels, badges, streaks)
- ☐ Milestone celebrations
- ☐ Challenge system
- ☐ Admin panel enhancement:
 - Template-based activity creation
 - Bulk import
 - Review queue
 - Analytics dashboard

Sprint 6 (Week 13): Notifications & Polish

- ☐ Implement all notification types
- ☐ Notification preferences UI
- ☐ UI/UX polish (animations, transitions)
- ☐ Performance optimization
- ☐ Accessibility improvements

Testing:

- ☐ Full regression testing
- ☐ Load testing (backend performance)
- ☐ Security audit (basic)
- ☐ Beta user testing (20-30 parents)

Beta Release:

- ☐ Deploy to production (soft launch)
- ☐ Closed beta group (invite-only)
- ☐ Collect structured feedback (surveys)
- ☐ Monitor analytics and errors

Deliverables:

- Feature-complete V1.0 (minus community features)
- 100+ activities
- Beta testing insights

5.4 Phase 3: V1.0 Launch (Week 14-16) - 3 weeks

Goal: Polish & public release

Week 14: QA & Fixes

- ☐ Address all beta feedback
- ☐ Fix critical bugs (P0/P1)
- ☐ Performance tuning (optimize slow queries)
- ☐ Security hardening
- ☐ Content expansion (target: 150 activities)

Week 15: Launch Prep

- ☐ App store submissions (metadata, screenshots, video)
- ☐ Privacy policy & terms of service (legal review)
- ☐ Help documentation & FAQ
- ☐ Onboarding tutorial refinement
- ☐ Marketing website (landing page)
- ☐ Social media setup (Instagram, Facebook, TikTok)
- ☐ Press kit (product description, images, team info)

Week 16: Launch Week

- ☐ App store approval received
- ☐ Final smoke testing on production
- ☐ Launch announcement (social media, email)
- ☐ PR outreach (parenting blogs, media)
- ☐ Monitor launch (support queries, crash rates)
- ☐ Hotfix readiness (team on standby)

Post-Launch (Weeks 17-18):

- ☐ Daily monitoring (analytics, errors, support)
- ☐ User feedback collection (in-app, email)
- ☐ Prioritize V1.1 backlog
- ☐ Retrospective meeting (what went well, what to improve)

Deliverables:

- Live on App Store & Play Store
- 150+ activities
- Marketing collateral
- Support channel active
- V1.1 roadmap defined

5.5 Phase 4: V1.5 - Community Features (Week 19-24) - 6 weeks

Goal: Build community & retention

Sprint 7 (Week 19-20): Activity Variations

- ☐ User submission flow (photo + note)
- ☐ Moderation system (review queue)
- ☐ Display variations on activity page
- ☐ "Helpful" upvote feature

Sprint 8 (Week 21-22): Recipe Exchange

- ☐ Recipe submission form (cooking activities)
- ☐ Uwa Farah team review workflow
- ☐ Recipe display (dedicated tab)
- ☐ Dietary filters (vegetarian, halal, etc.)

Sprint 9 (Week 23-24): Play Spots

- ☐ Location tagging on activity logs
- ☐ Play spot creation (crowd-sourced)
- ☐ Google Maps integration
- ☐ Play spot discovery page (map + list)
- ☐ Upvote/downvote system

Testing & Release:

- ☐ Community feature testing
- ☐ Moderation workflow testing
- ☐ Deploy V1.5 to production (phased rollout)

Deliverables:

- Active community features
- 500+ play spots (target by Month 6)
- User-generated content flowing

5.6 Phase 5: V2.0 Planning (Month 7+)

Potential Features (prioritize based on user feedback):

- ☐ Advanced ML recommendations
 - ☐ Child-facing interface (for older kids)
 - ☐ Parent groups (local communities)
 - ☐ In-app messaging (with heavy moderation)
 - ☐ Virtual workshops/events
 - ☐ Monetization (premium content, subscription tiers)
 - ☐ Multi-language support (English, regional languages)
 - ☐ School/educator partnerships
 - ☐ Offline mode
 - ☐ Wearable integration (smart watches for activity tracking)
-

6. Database Schema (Detailed)

sql

```
-- =====  
-- USERS & AUTHENTICATION  
-- =====
```

```
CREATE TABLE users (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  email VARCHAR(255) UNIQUE NOT NULL,  
  password_hash VARCHAR(255), -- nullable for OAuth users  
  full_name VARCHAR(255) NOT NULL,  
  username VARCHAR(50) UNIQUE, -- for community features  
  profile_photo_url TEXT,  
  location_city VARCHAR(100),
```

```
-- Gamification
```

```
level INTEGER DEFAULT 1,  
experience_points INTEGER DEFAULT 0,  
current_streak_days INTEGER DEFAULT 0,  
longest_streak_days INTEGER DEFAULT 0,  
last_activity_logged_at TIMESTAMP,
```

```
-- Preferences
```

```
notification_preferences JSONB DEFAULT '{  
  "scheduled_reminders": true,  
  "weekly_digest": true,  
  "milestone_celebrations": true,  
  "new_content_alerts": true,  
  "engagement_nudges": false  
}',
```

```
-- OAuth
```

```
google_id VARCHAR(255) UNIQUE,  
apple_id VARCHAR(255) UNIQUE,
```

```
-- Timestamps
```

```
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
last_login_at TIMESTAMP,
```

```
-- Soft delete
```

```
deleted_at TIMESTAMP,
```

```
-- Indexes
```

```
INDEX idx_email (email),
```

```
INDEX idx_username (username),
```

```
INDEX idx_created_at (created_at)
```

```

INDEX idx_created_at (created_at)
);

CREATE TABLE user_badges (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  badge_type VARCHAR(50) NOT NULL,
  earned_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  is_showcased BOOLEAN DEFAULT FALSE,

  UNIQUE(user_id, badge_type),
  INDEX idx_user_badges (user_id),
  INDEX idx_badge_type (badge_type)
);

```

```

-- =====
-- CHILDREN & PROFILES
-- =====

```

```

CREATE TABLE children (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  name VARCHAR(255) NOT NULL,
  birth_date DATE, -- nullable for prenatal
  expected_due_date DATE, -- nullable, for prenatal
  status VARCHAR(20) DEFAULT 'born', -- 'expecting' or 'born'
  profile_photo_url TEXT,
  gender VARCHAR(20), -- 'male', 'female', 'other', 'prefer_not_to_say'
  notes TEXT,

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  deleted_at TIMESTAMP,

  INDEX idx_user_children (user_id),
  INDEX idx_created_at (created_at)
);

```

```

-- =====
-- CONTENT LIBRARY
-- =====

```

```

CREATE TABLE activities (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  title VARCHAR(255) NOT NULL,
  slug VARCHAR(255) UNIQUE NOT NULL,
  description TEXT NOT NULL,

```


-- Classification

age_min_months INTEGER NOT NULL DEFAULT 0, -- 0 for prenatal
age_max_months INTEGER NOT NULL,
categories JSONB NOT NULL, -- ["Art", "Creativity"]
difficulty VARCHAR(20) DEFAULT 'easy', -- 'easy', 'medium', 'challenging'
estimated_duration_minutes INTEGER,
location_type VARCHAR(20), -- 'indoor', 'outdoor', 'both'

-- Content

materials JSONB, -- [{"name": "Paper", "quantity": "5 sheets", "is_common": true}]
steps JSONB, -- [{"order": 1, "instruction": "...", "image_url": "..."}]
educational_values JSONB, -- ["Creativity", "Fine Motor"]
parent_tips TEXT,
conversation_starters JSONB, -- ["What colors do you see?", "How does it feel?"]
safety_notes TEXT,

-- Media

main_image_url TEXT NOT NULL,
additional_images JSONB, -- ["url1", "url2"]
video_url TEXT,

-- Metadata

created_by UUID REFERENCES users(id),
status VARCHAR(20) DEFAULT 'draft', -- 'draft', 'pending_review', 'published', 'archived'

-- Analytics

popularity_score INTEGER DEFAULT 0,
view_count INTEGER DEFAULT 0,
completion_count INTEGER DEFAULT 0,
average_rating DECIMAL(3,2),

-- Timestamps

published_at TIMESTAMP,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

-- Indexes

INDEX idx_slug (slug),
INDEX idx_age_range (age_min_months, age_max_months),
INDEX idx_categories (categories) USING GIN,
INDEX idx_status (status),
INDEX idx_published_at (published_at),
INDEX idx_popularity (popularity_score DESC)

);

```

CREATE TABLE activity_templates (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name VARCHAR(255) NOT NULL,
  category VARCHAR(50),
  template_structure JSONB NOT NULL, -- Pre-defined sections
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- =====
-- ACTIVITY LOGGING & JOURNALING
-- =====

CREATE TABLE activity_logs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,
  activity_id UUID REFERENCES activities(id) ON DELETE SET NULL,
  custom_activity_title VARCHAR(255), -- if activity_id is null

  completed_at TIMESTAMP NOT NULL,

  -- Content
  photos JSONB, -- [{"url": "s3://...", "caption": "...", "order": 1}]
  notes TEXT,

  -- 4E Feedback
  feedback_4e JSONB, -- {"enjoy": true, "easy": false, "excellent": true, "earn": false}

  -- Additional
  duration_minutes INTEGER,
  location_name TEXT,
  location_spot_id UUID, -- FK to play_spots (future)
  mood_tags JSONB, -- ["happy", "curious", "frustrated"]
  is_favorite BOOLEAN DEFAULT FALSE,
  shared_as_variation BOOLEAN DEFAULT FALSE,

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  -- Indexes
  INDEX idx_child_logs (child_id),
  INDEX idx_activity_logs (activity_id),
  INDEX idx_completed_at (completed_at DESC)
);

CREATE TABLE scheduled_activities (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,

```

```

child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,
activity_id UUID NOT NULL REFERENCES activities(id) ON DELETE CASCADE,

scheduled_for TIMESTAMP NOT NULL,
reminder_before_minutes INTEGER DEFAULT 60,
reminder_sent BOOLEAN DEFAULT FALSE,
notes TEXT,

status VARCHAR(20) DEFAULT 'pending', -- 'pending', 'completed', 'cancelled'
completed_log_id UUID REFERENCES activity_logs(id),

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

INDEX idx_child_scheduled (child_id),
INDEX idx_scheduled_for (scheduled_for),
INDEX idx_status (status)
);

-- =====
-- INSIGHTS & ANALYTICS
-- =====

CREATE TABLE child_insights (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,

  insight_type VARCHAR(50) NOT NULL, -- 'talent_indicator', 'category_preference', 'skill_development'
  category VARCHAR(50), -- if category-specific
  message TEXT NOT NULL,
  confidence_level VARCHAR(20), -- 'low', 'medium', 'high'

  supporting_data JSONB, -- {"activities_count": 15, "enjoy_rate": 0.93}

  generated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  is_viewed BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  INDEX idx_child_insights (child_id),
  INDEX idx_insight_type (insight_type),
  INDEX idx_generated_at (generated_at DESC)
);

-- =====
-- COMMUNITY FEATURES
-- =====

```

```
CREATE TABLE play_spots (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  name VARCHAR(255) NOT NULL,  
  type VARCHAR(50), -- 'playground', 'park', 'indoor_play', 'library', 'museum'  
  address TEXT,  
  latitude DECIMAL(10, 8),  
  longitude DECIMAL(11, 8),  
  
  age_suitability_min INTEGER,  
  age_suitability_max INTEGER,  
  facilities JSONB, -- ["parking", "restroom", "food_stall"]  
  
  photos JSONB,  
  notes TEXT,  
  
  created_by UUID REFERENCES users(id),  
  upvotes INTEGER DEFAULT 0,  
  downvotes INTEGER DEFAULT 0,  
  activity_count INTEGER DEFAULT 0,  
  
  status VARCHAR(20) DEFAULT 'active', -- 'active', 'flagged', 'removed'  
  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
  INDEX idx_location (latitude, longitude),  
  INDEX idx_type (type),  
  INDEX idx_created_by (created_by)  
);
```

```
CREATE TABLE activity_variations (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  parent_activity_id UUID NOT NULL REFERENCES activities(id) ON DELETE CASCADE,  
  submitted_by UUID NOT NULL REFERENCES users(id),  
  
  variation_title VARCHAR(255),  
  variation_notes TEXT, -- max 500 chars  
  photo_url TEXT,  
  
  helpful_count INTEGER DEFAULT 0,  
  
  status VARCHAR(20) DEFAULT 'pending', -- 'pending', 'approved', 'rejected'  
  reviewed_by UUID REFERENCES users(id),  
  reviewed_at TIMESTAMP,  
  review_notes TEXT,  
  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
INDEX idx_parent_activity (parent_activity_id),  
INDEX idx_status (status)  
);
```

```
CREATE TABLE parent_recipes (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  parent_activity_id UUID NOT NULL REFERENCES activities(id) ON DELETE CASCADE,  
  submitted_by UUID NOT NULL REFERENCES users(id),  
  
  recipe_title VARCHAR(255) NOT NULL,  
  ingredients JSONB, -- [{"item": "Banana", "quantity": "2"}]  
  steps JSONB,  
  prep_time_minutes INTEGER,  
  servings VARCHAR(50),  
  dietary_tags JSONB, -- ["vegetarian", "halal", "nut-free"]  
  
  photo_url TEXT,  
  notes TEXT,  
  helpful_count INTEGER DEFAULT 0,  
  
  status VARCHAR(20) DEFAULT 'pending',  
  reviewed_by UUID REFERENCES users(id),  
  review_notes TEXT,  
  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
  INDEX idx_parent_activity (parent_activity_id),  
  INDEX idx_status (status)  
);
```

```
CREATE TABLE parent_tips (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  submitted_by UUID NOT NULL REFERENCES users(id),  
  
  category VARCHAR(50), -- 'activity_hacks', 'nutrition', 'sleep', 'books', 'general'  
  title VARCHAR(100) NOT NULL,  
  content TEXT NOT NULL, -- max 1000 chars  
  photo_url TEXT,  
  tags JSONB,  
  
  helpful_count INTEGER DEFAULT 0,  
  
  status VARCHAR(20) DEFAULT 'pending', -- 'pending', 'approved', 'rejected', 'featured'  
  reviewed_by UUID REFERENCES users(id),
```

```

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

INDEX idx_category (category),
INDEX idx_status (status),
INDEX idx_created_at (created_at DESC)
);

-- =====
-- NOTIFICATIONS & ENGAGEMENT
-- =====

CREATE TABLE notifications (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

  type VARCHAR(50) NOT NULL, -- 'scheduled_reminder', 'milestone', 'badge_earned', etc.
  title VARCHAR(255) NOT NULL,
  message TEXT NOT NULL,
  data JSONB, -- Context-specific payload

  is_read BOOLEAN DEFAULT FALSE,
  sent_at TIMESTAMP,
  scheduled_for TIMESTAMP,
  delivery_status VARCHAR(20), -- 'pending', 'sent', 'failed'

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  INDEX idx_user_notifications (user_id),
  INDEX idx_type (type),
  INDEX idx_scheduled_for (scheduled_for)
);

CREATE TABLE push_notification_tokens (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  device_token TEXT NOT NULL,
  platform VARCHAR(20), -- 'android', 'ios'
  is_active BOOLEAN DEFAULT TRUE,

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  UNIQUE(user_id, device_token),
  INDEX idx_user_tokens (user_id)
);

```

```
-- =====  
-- GAMIFICATION  
-- =====
```

```
CREATE TABLE challenges (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  title VARCHAR(255) NOT NULL,  
  description TEXT,  
  
  challenge_type VARCHAR(50), -- 'category_specific', 'diversity', 'consistency', 'community'  
  start_date DATE NOT NULL,  
  end_date DATE NOT NULL,  
  
  goal_config JSONB, -- {"target_count": 5, "category": "Outdoor"}  
  reward_badge_type VARCHAR(50),  
  reward_xp INTEGER,  
  
  is_active BOOLEAN DEFAULT TRUE,  
  participant_count INTEGER DEFAULT 0,  
  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
  INDEX idx_dates (start_date, end_date),  
  INDEX idx_active (is_active)  
);
```

```
CREATE TABLE challenge_participants (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  challenge_id UUID NOT NULL REFERENCES challenges(id) ON DELETE CASCADE,  
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,  
  
  progress INTEGER DEFAULT 0,  
  completed BOOLEAN DEFAULT FALSE,  
  completed_at TIMESTAMP,  
  
  joined_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
  UNIQUE(challenge_id, user_id),  
  INDEX idx_challenge (challenge_id),  
  INDEX idx_user (user_id)  
);
```

```
CREATE TABLE child_milestones (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
```

```

child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,

milestone_type VARCHAR(50) NOT NULL, -- 'first_activity', 'ten_activities', 'found_favorite'
category VARCHAR(50),

achieved_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
is_celebrated BOOLEAN DEFAULT FALSE,

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

INDEX idx_child_milestones (child_id),
INDEX idx_milestone_type (milestone_type)
);

-- =====
-- ADMIN & CONTENT MANAGEMENT
-- =====

CREATE TABLE content_review_queue (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  content_type VARCHAR(50) NOT NULL, -- 'activity', 'recipe', 'tip', 'play_spot'
  content_id UUID NOT NULL, -- Polymorphic reference

  submitted_by UUID REFERENCES users(id),
  status VARCHAR(20) DEFAULT 'pending', -- 'pending', 'in_review', 'approved', 'rejected'

  reviewer_id UUID REFERENCES users(id),
  review_notes TEXT,
  priority VARCHAR(20) DEFAULT 'normal', -- 'low', 'normal', 'high'

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  reviewed_at TIMESTAMP,

  INDEX idx_status (status),
  INDEX idx_content_type (content_type),
  INDEX idx_created_at (created_at)
);

CREATE TABLE admin_activity_log (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  admin_user_id UUID NOT NULL REFERENCES users(id),

  action VARCHAR(100) NOT NULL, -- 'create_activity', 'approve_content', 'publish_challenge'
  target_type VARCHAR(50),
  target_id UUID,
  metadata JSONB,

```



```
ip_address INET,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
INDEX idx_admin (admin_user_id),  
INDEX idx_action (action),  
INDEX idx_created_at (created_at DESC)  
);
```

7. Development Methodology

7.1 Agile Scrum Approach

Sprint Duration: 2 weeks

Team Roles:

- **Product Owner:** Project lead (defines priorities, accepts stories)
- **Scrum Master:** Technical validator (facilitates, removes blockers)
- **Development Team:** AI + Human validators
- **Stakeholders:** Uwa Farah (content validation), Psikolog team (educational validation)

Ceremonies:

- **Sprint Planning** (Monday Week 1, 2 hrs): Select backlog items, define sprint goal
- **Daily Standup** (15 min): What did yesterday, today, blockers (async via Slack if remote)
- **Sprint Review** (Friday Week 2, 1 hr): Demo completed features to stakeholders
- **Sprint Retrospective** (Friday Week 2, 1 hr): What went well, what to improve

7.2 AI-Assisted Development Workflow

AI Tools: OpenAI Codex (via GitHub Copilot) + Anthropic Claude

Workflow:

1. **Task Definition** (Human):

- Write detailed user story with acceptance criteria
- Example: "As a parent, I want to log an activity with photos so that I can preserve memories."

2. **Context Preparation** (Human):

- Gather relevant context (existing code, data models, API contracts)
- Write clear prompt with requirements

3. **Code Generation** (AI):

- Use Copilot in IDE for in-line suggestions
- OR: Paste context + requirements to Claude for full function/component generation

4. **Code Review** (Human):

- Validate logic correctness
- Check edge cases and error handling
- Ensure adherence to coding standards
- Refactor if needed for readability

5. **Testing** (Human + AI):

- AI can generate unit test scaffolds
- Human writes assertions and edge case tests
- Run tests, iterate until passing

6. **Integration** (Human):

- Merge into codebase
- Commit with descriptive message
- Create pull request for team review (if applicable)

7. **Documentation** (Human + AI):

- Add code comments (AI can generate docstrings)
- Update README or technical docs (AI can draft, human refines)

Best Practices:

- Start with small, well-defined tasks for AI
- Always validate AI output (don't blindly accept)
- Iterate: If AI output isn't right, refine prompt and retry
- Learn from AI: Understand *why* it wrote code a certain way
- Capture good prompts in a "prompt library" for reuse

7.3 Code Quality Standards

Linting & Formatting:

- **Flutter:** `flutter analyze`, `dartfmt`
- **Backend:** ESLint + Prettier (TypeScript)
- Pre-commit hooks: Auto-format and lint before commit (using Husky)

Code Review:

- All code must be reviewed (even if AI-generated)
- Checklist:
 - ☐ Functionality works as expected
 - ☐ Edge cases handled
 - ☐ Error handling present
 - ☐ No security vulnerabilities
 - ☐ Tests written and passing
 - ☐ Code is readable and well-commented
 - ☐ Follows project conventions

Testing Requirements:

- **Unit Tests:** All business logic functions (target: 70%+ coverage)
- **Integration Tests:** API endpoints (happy path + error cases)
- **Widget Tests:** Flutter screens with critical interactions
- **E2E Tests:** Key user flows (login → browse → log activity) - optional, but recommended

Documentation Standards:

- README in each major directory (purpose, setup instructions)
- API docs: OpenAPI/Swagger spec (auto-generated from code)
- Code comments: Why, not just what (especially for complex logic)
- Changelog: Keep CHANGELOG.md updated with each release

7.4 Version Control & Branching

Branching Strategy: Git Flow

Branches:

- `main`: Production-ready code (always deployable)
- `develop`: Integration branch for features
- `feature/*`: Individual features (e.g., `feature/activity-logging`)
- `bugfix/*`: Bug fixes
- `hotfix/*`: Urgent production fixes

Workflow:

1. Create feature branch from `develop`
2. Develop feature (commit often with clear messages)
3. Open pull request to `develop`
4. Code review + CI checks pass
5. Merge to `develop`
6. At sprint end: Merge `develop` to `main` + tag release (v1.0.0)

Commit Messages: Conventional Commits format

```
feat: add activity scheduling feature
fix: resolve photo upload timeout issue
docs: update API documentation for logging endpoint
test: add unit tests for recommendation algorithm
```

7.5 Testing Strategy

Test Pyramid:

```
  ^
 / \ E2E (few)
/----\
/ Integration \
/ (moderate)  \
/-----\
/ Unit (many)  \
/-----\
```

Testing Tools:

- **Flutter:** flutter_test, mockito (mocking)
- **Backend:** Jest (unit & integration), supertest (HTTP testing)

Test Coverage Goals:

- Unit tests: 70%+ coverage (focus on business logic)
- Integration tests: Cover all API endpoints (happy path minimum)
- E2E tests: Cover 3-5 critical user journeys

Continuous Testing:

- CI runs all tests on every push
- Pull requests blocked if tests fail
- Weekly coverage report review

8. Content Strategy

8.1 Content Library Target

Launch (V1.0): 150 activities

- Prenatal: 10 activities (bonding, preparation, wellness)
- 0-1 year: 20 activities (sensory, tummy time, etc.)
- 1-3 years: 25 activities (motor skills, exploration)
- 3-5 years: 30 activities (creativity, pre-literacy)
- 6-8 years: 25 activities (STEM, sports, reading)
- 9-11 years: 20 activities (hobbies, projects)
- 12-15 years: 20 activities (advanced, community service)

Categories Distribution:

- Art & Creativity: 25 activities
- Science & Discovery: 20 activities
- Physical & Outdoor: 25 activities
- Literacy & Language: 20 activities
- Music & Rhythm: 15 activities
- Cooking & Nutrition: 15 activities
- Social & Emotional: 15 activities
- Cognitive & Logic: 15 activities

8.2 Content Creation Process

Roles:

- **Content Creator:** Uwa Farah's team (5-7 people)
- **Reviewer:** Uwa Farah + 1 senior educator
- **Approver:** Psikolog team (monthly batch review)

Workflow:

1. **Ideation** (Weekly meeting): Brainstorm 10-15 activity ideas
2. **Assignment**: Distribute to content creators
3. **Creation** (3-5 days): Use templates in admin panel to draft activities
4. **Internal Review** (2 days): Uwa Farah reviews drafts, provides feedback
5. **Revision** (1 day): Creators address feedback
6. **Approval**: Uwa Farah approves → Status: Published
7. **Monthly Quality Review**: Psikolog team reviews all published content (audit)

Content Creation Targets:

- Pre-launch: 150 activities (6 weeks, team of 6 = 4 activities/person/week)
- Post-launch: 8-10 new activities/week (ongoing)

8.3 Content Guidelines

Activity Design Principles:

- **Age-appropriate**: Match cognitive and physical capabilities
- **Safe**: Clear safety notes for any risks
- **Accessible**: Use common household materials when possible
- **Engaging**: Fun, not just educational (balance learning and joy)
- **Inclusive**: Adaptable for different abilities and backgrounds
- **Parent-friendly**: Clear instructions, realistic time estimates

Quality Checklist:

- ☐ Title is descriptive and engaging
- ☐ Description explains what and why (150 chars)
- ☐ Age range is appropriate
- ☐ Materials list is complete and specific
- ☐ Steps are clear, numbered, and actionable
- ☐ Educational value is explicitly stated
- ☐ Parent tips provide helpful context
- ☐ Photos/videos are high quality and relevant
- ☐ Safety notes included if applicable
- ☐ No spelling/grammar errors

Content Variety:

- Mix of activity types: Quick (15min), medium (30-60min), extended (2hr+)
- Mix of difficulty: Easy (minimal prep), Medium (some prep), Challenging (complex)
- Mix of settings: Indoor, outdoor, both
- Mix of seasons: Activities for all weather/seasons
- Cultural relevance: Include Indonesian traditions, holidays, local context

8.4 Content Maintenance

Quarterly Content Audit:

- Review activity performance (completion rate, ratings)
- Retire underperforming activities (move to archive, not delete)
- Refresh top performers with new variations
- Update seasonal content (before season starts)

User Feedback Integration:

- Monitor activity ratings and user comments
- Surface issues to content team (unclear instructions, safety concerns)
- Iterate activities based on feedback

Trending Content:

- Spotlight popular activities in weekly digest
 - Create "related activities" based on user completion patterns
 - Feature seasonal/holiday activities prominently
-

9. Go-to-Market Strategy

9.1 Target Market Segmentation

Primary Market: Urban Indonesia (Jakarta, Surabaya, Bandung, Medan, Bali)

- Population: ~50 million in target cities
- Smartphone penetration: 70%+
- Target segment: Parents with children 0-15 years (estimated 10 million)

Initial Target: "Early Adopters" (Year 1)

- Tech-savvy parents (use multiple apps daily)
- Education-conscious (invest in child development)
- Middle-upper income (can afford activities and smartphone)
- Estimated reach: 50,000 users in first 12 months

9.2 Value Proposition

For Parents:

- "One app to track your child's entire journey from pregnancy to teenage years"
- "Never run out of activity ideas - personalized recommendations every day"
- "Discover your child's talents early with AI-powered insights"
- "Preserve precious moments with beautifully organized timelines"
- "Join a community of supportive parents sharing tips and experiences"

Differentiation (vs. generic parenting apps):

- **Complete coverage:** Prenatal to teen (most apps focus on 0-5)
- **Curated content:** Expert-validated activities (not just Pinterest reposts)
- **Intelligence:** Talent discovery via 4E framework + pattern recognition
- **Gamification:** Makes parenting engagement fun and rewarding
- **Local:** Indonesian context, language, and community

9.3 Launch Strategy

Phase 1: Closed Beta (Weeks 14-15)

- Invite 100 parents (friends, family, parenting groups)
- Goal: Collect feedback, identify bugs, generate early content (user logs)
- Incentive: "Founding Member" badge, lifetime premium (if future paid features)

Phase 2: Soft Launch (Week 16)

- Open registration (no invites needed)
- Limited marketing: Social media posts, parenting Facebook groups
- Goal: Validate server load, refine onboarding, build word-of-mouth
- Target: 500 users in first week

Phase 3: Public Launch (Week 17-18)

- Press release to parenting media, tech blogs
- App Store / Play Store feature submission (editorial team consideration)
- Social media campaign (#JejakAnakChallenge - share child's first activity)
- Influencer partnerships (parenting bloggers, Instagram moms)
- Goal: 5,000 users in first month

Phase 4: Growth (Month 2-12)

- Content marketing: Weekly blog posts, parenting tips
- SEO: Optimize website for "kegiatan anak", "ide bermain anak", etc.
- App store optimization: Keywords, screenshots, reviews
- Referral program: "Invite a friend, both get XP bonus"
- Partnerships: Collaborate with preschools, parenting workshops
- Community events: Monthly virtual parenting Q&A with Uwa Farah

9.4 Marketing Channels

Digital:

- **Instagram:** Visual platform (showcase activities with photos)
 - Daily activity ideas
 - Parent testimonials
 - Behind-the-scenes content creation
- **Facebook Groups:** Join/create parenting communities
- **TikTok:** Short-form videos (15-sec activity demos)
- **YouTube:** Long-form content (activity tutorials, parenting tips)
- **Google Ads:** Search campaigns ("aplikasi parenting", "ide main anak")
- **Facebook/Instagram Ads:** Targeted ads (parents aged 25-45, Indonesia)

Content Marketing:

- SEO-optimized blog on website
- Guest posts on popular parenting blogs
- Email newsletter (weekly digest to registered users)

Partnerships:

- Preschools/kindergartens: Recommend app to parents
- Parenting coaches: Co-branded workshops
- Pediatricians: Referral program (trusted recommendation)
- Brand collaborations: Sponsor activities (e.g., "Activity by [Brand]")

Community:

- User-generated content campaigns
- Monthly challenges with prizes (best activity log, most creative variation)
- Feature "Parent of the Month" in app and social media

9.5 Monetization Strategy (Future - V2)

Freemium Model:

- **Free Tier (V1):**
 - Unlimited activity logging
 - Access to 80% of activity library
 - Basic recommendations
 - Community features
- **Premium Tier (V2):**
 - Unlock all activities (including exclusive expert content)
 - Advanced AI recommendations
 - Detailed insights & progress reports
 - Ad-free experience
 - Priority support
 - Export timeline as PDF book (print-ready)
 - Pricing: Rp 49,000/month or Rp 399,000/year (~\$3-25 USD)

Other Revenue Streams (V3+):

- **B2B:** License content to schools/educators
- **Sponsored Activities:** Brands pay to feature activities using their products
- **Events:** Paid virtual workshops with experts
- **Marketplace:** Commission from activity materials sold through partnerships

Note: V1.0 is 100% free (no paywalls) to maximize adoption and feedback.

10. Success Metrics & KPIs

10.1 North Star Metric

Activities Logged Per Week (Across All Users)

- **Rationale:** Core value loop is logging activities. More logs = more engagement = app is useful
- **Target:** 1,000 activities/week by Month 3, 5,000/week by Month 12

10.2 Acquisition Metrics

- **Downloads:** App Store + Play Store installs
 - **Target:** 10,000 downloads by Month 3, 75,000 by Month 12
- **Registrations:** Completed signups
 - **Target:** 60% of downloads convert to registration
- **Onboarding Completion:** Users who add first child profile
 - **Target:** 80% of registrations complete onboarding

10.3 Engagement Metrics

- **Daily Active Users (DAU):** Users who open app daily
 - Target: 15% of MAU
- **Monthly Active Users (MAU):** Users who open app in 30 days
 - Target: 25,000 by Month 12
- **Activities per User per Week:** Average logs per active user
 - Target: 3 activities/user/week
- **Session Duration:** Time spent in app per session
 - Target: 5+ minutes (indicates exploration, not just quick log)
- **Feature Adoption:**
 - % users who schedule activities: Target 40%
 - % users who check insights: Target 60%
 - % users who participate in challenges: Target 25%

10.4 Retention Metrics

- **Day 1 Retention:** % users who return next day
 - Target: 50%
- **Day 7 Retention:** % users who return after 1 week
 - Target: 40%
- **Day 30 Retention:** % users who return after 1 month
 - Target: 35%
- **Churn Rate:** % users who stop using (no activity in 60 days)
 - Target: <5% monthly

10.5 Content Metrics

- **Activity Completion Rate:** % of activities marked as done after viewing
 - Target: 25% (realistic - not all views lead to doing activity)
- **Average Activity Rating:** User satisfaction (future feature)
 - Target: 4.2/5.0
- **Content Freshness:** New activities added per week
 - Target: 8-10/week post-launch
- **Category Coverage:** Activities available per category per age group
 - Target: Min 10 activities per category per age group

10.6 Community Metrics

- **User Contributions:** Variations, recipes, tips, play spots submitted
 - Target: 500 submissions by Month 6
- **Contribution Approval Rate:** % of submissions approved
 - Target: 70% (indicates good community culture)
- **Helpful Votes:** Engagement with community content
 - Target: Avg 10 helpful votes per approved contribution

10.7 Quality Metrics

- **App Crashes:** Crash-free sessions
 - Target: >99.5% (industry standard)
- **API Response Time:** P95 latency
 - Target: <500ms
- **App Store Rating:** Public ratings
 - Target: 4.5+ stars (iOS and Android)
- **Net Promoter Score (NPS):** Would recommend app?
 - Target: 50+ (good), 70+ (excellent)

10.8 Business Metrics (Future)

- **Cost per Acquisition (CPA):** Marketing spend / new users
 - Target: <Rp 10,000 per user
- **Customer Lifetime Value (LTV):** If premium model
 - Target: LTV > 3x CPA
- **Premium Conversion Rate:** % free users who upgrade
 - Target: 5-10% (industry standard for freemium)

10.9 Monitoring & Reporting

Tools:

- Firebase Analytics + Google Analytics 4 (user behavior)
- Mixpanel or Amplitude (funnel analysis, retention cohorts)
- Grafana dashboards (real-time tech metrics)
- Weekly automated report (email to team)

Review Cadence:

- Daily: Crash rate, critical errors
 - Weekly: Engagement metrics, feature adoption
 - Monthly: Retention, content performance, NPS
 - Quarterly: Strategic review (adjust roadmap based on data)
-

11. Risk Management

11.1 Technical Risks

Risk: Scalability issues during user growth spike

- **Mitigation:** Load testing before launch, auto-scaling infrastructure (ECS), database query optimization
- **Contingency:** Temporarily throttle new registrations if needed

Risk: Data loss or corruption

- **Mitigation:** Automated daily backups (RDS), point-in-time recovery, multi-AZ database
- **Contingency:** Disaster recovery plan documented, regular restore tests

Risk: Security breach (unauthorized access to child data)

- **Mitigation:** Security best practices (encryption, auth, input validation), regular security audits, bug bounty program (future)
- **Contingency:** Incident response plan, immediate user notification if breach occurs

Risk: Third-party service outage (AWS, FCM)

- **Mitigation:** Multi-region infrastructure (if budget allows), fallback mechanisms (graceful degradation)
- **Contingency:** Status page for users, transparent communication

11.2 Product Risks

Risk: Low user adoption (product-market fit not validated)

- **Mitigation:** Extensive prototype testing, beta user feedback, iterate on UX
- **Contingency:** Pivot strategy, re-evaluate value proposition, user interviews

Risk: Content quality issues (inaccurate or unsafe activities)

- **Mitigation:** Expert review process, user flagging system, quarterly audits
- **Contingency:** Immediate content removal, publish correction, review process improvement

Risk: Poor user retention (users download but don't engage)

- **Mitigation:** Onboarding optimization, push notifications, gamification
- **Contingency:** User surveys (why churning?), feature adjustments, re-engagement campaigns

Risk: Community toxicity (inappropriate user-generated content)

- **Mitigation:** Pre-moderation, clear guidelines, report/ban system
- **Contingency:** Increase moderation resources, tighten approval criteria

11.3 Team Risks

Risk: AI-generated code contains critical bugs

- **Mitigation:** Mandatory code review, comprehensive testing, human oversight
- **Contingency:** Rollback capability, hotfix process

Risk: Content team overwhelmed (can't produce enough activities)

- **Mitigation:** Start with achievable targets, provide efficient tools (templates, bulk import)
- **Contingency:** Hire additional content creators, adjust content cadence (quality over quantity)

Risk: Key team member leaves (knowledge loss)

- **Mitigation:** Thorough documentation, knowledge sharing sessions, cross-training
- **Contingency:** Hire replacement quickly, leverage docs to onboard

11.4 Legal & Compliance Risks

Risk: Privacy law violations (GDPR, child data protection)

- **Mitigation:** Legal review of privacy policy, implement data controls (export, deletion), minimize data collection
- **Contingency:** Legal counsel on retainer, rapid compliance fixes

Risk: Copyright issues (user uploads copyrighted content)

- **Mitigation:** Terms of service (user owns content, grants license), DMCA process
- **Contingency:** Remove infringing content promptly, ban repeat offenders

Risk: Liability for activity injuries (child hurt following activity)

- **Mitigation:** Clear disclaimers ("adult supervision required"), safety notes in activities, terms of service (use at own risk)
- **Contingency:** Legal insurance, incident reporting process

11.5 Market Risks

Risk: Competitor launches similar app

- **Mitigation:** Move fast, build moat (community, content quality, brand), unique features (4E framework, prenatal support)
- **Contingency:** Differentiate further, emphasize local expertise (Indonesian context)

Risk: Market too small (not enough parents want this)

- **Mitigation:** Validate with prototype, beta testing, market research
 - **Contingency:** Expand target market (e.g., Southeast Asia), pivot to B2B (sell to schools)
-

12. Post-Launch Plan

12.1 First 30 Days Post-Launch

Week 1:

- Monitor daily: Crash reports, critical bugs, user support inquiries
- Emergency hotfixes if needed (deploy within 24 hours)
- Social media engagement: Respond to all comments, share user content
- Internal retrospective: What went well in launch, what to improve

Week 2-3:

- Analyze onboarding funnel: Where are users dropping off?
- A/B test improvements (e.g., different onboarding copy)
- Collect qualitative feedback: In-app surveys, email to users
- Content sprint: Add 10 more activities based on early requests

Week 4:

- First feature update (V1.1 planning based on feedback)
- User appreciation: Feature "User of the Week" in app/social
- Iterate notification strategy (if open rates low)
- Prepare growth experiments (referral incentives, content marketing)

12.2 Continuous Improvement Cycle

Monthly:

- Release feature update (new features, improvements, bug fixes)
- Review metrics dashboard (compare to targets, identify trends)
- User feedback synthesis (categorize, prioritize requests)
- Content team workshop (review performance, plan next month's activities)

Quarterly:

- Strategic planning session (review OKRs, adjust roadmap)
- Major feature launches (V1.5, V2.0 milestones)
- Team retrospective (process improvements, celebrate wins)
- Financial review (costs, revenue if applicable, budget adjustment)

12.3 User Support

Channels:

- **In-App:** Help center with FAQs, contact form
- **Email:** support@jejakanak.app (target: <24hr response time)
- **Social Media:** Instagram/Facebook DMs (daily monitoring)

Support Tiers:

- **Tier 1: Self-Service** (80% of queries):
 - Comprehensive FAQ
 - Video tutorials
 - Troubleshooting guides
- **Tier 2: Automated** (15% of queries):
 - Chatbot for common questions (future)
 - Email auto-responder with helpful links
- **Tier 3: Human Support** (5% of queries):
 - Tech team or dedicated support person
 - Response SLA: Critical (4 hrs), High (24 hrs), Normal (48 hrs)

Escalation:

- User reports child safety issue → Immediate escalation to senior team + legal review
- User reports bug causing data loss → Immediate escalation to tech lead

12.4 Community Management

Moderation Team:

- Start: 1 part-time moderator (Uwa Farah team member)
- Scale: Add moderators as community grows (1 moderator per 5,000 active users)

Moderation Guidelines:

- Remove content: Spam, harmful, inappropriate, off-topic
- Warn users: First offense (friendly reminder of guidelines)
- Ban users: Repeat offenders, severe violations (child safety)

Community Engagement:

- Highlight great contributions (weekly feature)
- Host virtual events (monthly parenting Q&A)
- Recognize top contributors (badges, special privileges)

13. Conclusion & Next Steps

13.1 Summary

JejakAnak is positioned to become the leading child development companion app in Indonesia by offering:

- **Comprehensive Coverage:** Prenatal to teenage years (unique in market)
- **Expert-Curated Content:** Quality activities validated by educators and psychologists
- **Intelligent Insights:** Talent discovery through 4E framework and pattern recognition
- **Engaging Experience:** Gamification and community features drive retention
- **Local Relevance:** Indonesian context, language, and community

With a robust technical foundation (Flutter + NestJS + PostgreSQL + AWS), modular architecture for scalability, and AI-assisted development for speed, the team is well-equipped to execute on this vision.

13.2 Immediate Next Steps

1. **Stakeholder Sign-Off** (Day 1):

- Review this masterplan with full team
- Address questions, confirm alignment
- Commit to timeline and resource allocation

2. **Kick Off Phase 0** (Day 2-3):

- Assign roles (who does what)
- Set up project management tools (Notion, Jira, or similar)
- Schedule design sprint (Week 1)

3. **Begin Design Sprint** (Week 1):

- Finalize user stories
- Uwa Farah team: Start ideating seed activities
- Design team (or designer): Begin Figma mockups

4. **Build Prototype** (Week 2-3):

- Develop dummy web app (interactive prototype)
- Parallel: Content team prepares 30 activities in spreadsheet
- Schedule validation sessions with psikolog and parents

5. **Validate & Iterate** (Week 3):








- Conduct user testing
- Refine design based on feedback
- Get final approval to proceed to development

6. **Start Development** (Week 4):

- Initialize GitHub repos
- Set up development environments
- Begin Sprint 1 (Foundation)

13.3 Success Criteria for V1.0

By end of Phase 3 (Week 16), the team will have:

-  Fully functional mobile app on iOS and Android
-  150+ curated activities in library
-  1,000+ registered users (beta + soft launch)
-  Core features complete (browse, schedule, log, insights, gamification)
-  Stable infrastructure (99.5%+ uptime)
-  Positive user feedback (NPS > 40)
-  Clear roadmap for V1.5 (community features)

If these criteria are met, JejakAnak V1.0 will be a validated product ready for growth and scale.

13.4 Long-Term Vision (3-5 Years)

Year 1: Establish product-market fit in Indonesia

- 50,000 active users
- 300+ activities
- Strong community engagement

Year 2-3: Scale and monetize

- 250,000+ active users
- Premium tier launch (freemium model)
- Expand to Southeast Asia (Malaysia, Singapore, Philippines)
- B2B partnerships (schools, preschools)

Year 4-5: Platform and ecosystem

- 1M+ active users
- Multi-language support (English, Bahasa Melayu, Tagalog)
- Educator marketplace (third-party content creators)
- Hardware integration (smart watches, IoT toys)
- Recognized as the leading parenting platform in region

Ultimate Goal: Empower every parent to raise children who reach their full potential, with JejakAnak as their trusted companion from pregnancy through adolescence. Every child deserves to have their unique journey documented, celebrated, and optimized for growth.

Document Version: 2.0 (Final)

Last Updated: October 20, 2025

Owner: JejakAnak Product Team

Status: Approved for Development

Appendices

Appendix A: Glossary

- **4E Framework:** Talent assessment method (Enjoy, Easy, Excellent, Earn)
- **Activity Log:** User-generated record of completed activity
- **Activity Variation:** User-submitted modification or alternative approach to official activity
- **Admin Panel:** Web interface for content management
- **Badge:** Achievement reward in gamification system
- **Child Profile:** Data record for one child within parent account
- **DAU/MAU:** Daily/Monthly Active Users
- **Dummy Prototype:** Interactive web prototype for validation
- **ECS:** Elastic Container Service (AWS)
- **FCM:** Firebase Cloud Messaging (push notifications)
- **Gamification:** Game-like elements (XP, badges, streaks) to drive engagement
- **Insight:** AI-generated observation about child's preferences or talents
- **JWT:** JSON Web Token (authentication method)
- **NPS:** Net Promoter Score (customer satisfaction metric)
- **ORM:** Object-Relational Mapping (database abstraction)
- **Play Spot:** User-contributed location for child activities
- **Prenatal:** Period before child birth (pregnancy phase)
- **RDS:** Relational Database Service (AWS managed PostgreSQL)
- **S3:** Simple Storage Service (AWS file storage)
- **Seed Activities:** Initial content populated before launch
- **Streak:** Consecutive days of activity logging

Appendix B: Contact Information

- **Product Owner:** [Name] - [email]
- **Technical Lead:** [Name] - [email]
- **Content Validator:** Uwa Farah - [email]
- **Psikolog Team Lead:** [Name] - [email]
- **Support Email:** support@jejakanak.app
- **GitHub:** github.com/jejakanak/app (private repo)
- **Figma Design:** [Link to Figma project]
- **Project Management:** [Link to Notion/Jira]

Appendix C: References

1. 4E Talent Framework: [Jejak Anak Kita (JENAKA) methodology]
2. Child Development Guidelines: WHO Child Development Standards
3. Flutter Documentation: flutter.dev
4. NestJS Documentation: nestjs.com
5. PostgreSQL Best Practices: postgresql.org
6. AWS Well-Architected Framework: aws.amazon.com/architecture
7. GDPR Compliance: gdpr.eu
8. App Store Optimization: Apple/Google developer guidelines

END OF MASTERPLAN V2.0