

SomaSmart - Offline generated AI tutor

Functions of the App



1. Delivering Personalized Learning Experiences

- Use AI to adapt lessons based on student performance and progress.
- Provide tailored recommendations (e.g., repeat this lesson, try this quiz).
- Adjust difficulty levels or pacing based on the learner's needs.



2. Operate Offline with Periodic Sync

- Store all essential content (lessons, quizzes, media) locally on the device.
- Work without an active internet connection.
- Sync data (progress, new lessons, payments) when internet becomes available.



3. Support Multiple User Roles

For Students:

- Allow access to lessons, practice exercises, and quizzes.
- Show progress tracking, badges, and completion status.

For Teachers:

- Enable uploading of lessons, tagging them by subject/level.
- Allow lesson previews and feedback from admins.

For Admins:

- Provide dashboard for user management, content review, payment tracking.
- For Schools:
 - Let schools create and manage teacher/student accounts.
 - Show engagement and performance analytics.



4. Manage Educational Content

- Host lessons categorized by level (Preschool, Primary, Secondary) and subject.
- Allow multimedia content: text, images, audio, video, interactive quizzes.
- Enable continuous updates to lessons through teacher uploads or admin updates.



5. Track and Report Progress

- Record student activity: complete lessons, quiz scores, usage time.
- Provide insights to students, teachers, and school admins.
- Offer downloadable or syncable performance reports.



6. Handle Secure User Access & Data Management

- Allow secure login for students, teachers, and admins.
- Store user data securely offline and sync it securely when online.
- Handle student privacy and parental controls responsibly.



7. Manage Payments & Access Control

- Handle mobile money, coupon codes, vouchers, or agent payments.
- Unlock content based on payment method or school license.

- Track payment status and access validity per user.

8. Provide Feedback & Assistance

- Give instant feedback on quizzes (correct/incorrect, explanations).
- Use AI to explain why an answer is wrong and suggest the next steps.
- Include help or tutorial sections for users.

9. Support Multiple Languages and Local Curriculum

- Localize content to match national curriculum.
- Allow lessons in English, Swahili, or other local languages.
- Support teacher-generated content aligned to regional standards.

10. Generate Analytics & Reports for Stakeholders

- Summary reports for:
 - Students (self-progress)
 - Teachers (class/student performance)
 - Schools (overall usage)
 - Admins (system-wide metrics)
- Use reports to refine teaching and support school planning.

11. Facilitate Communication (Optional)

- Allow offline feedback between students and teachers (saved locally and synced).
- Push messages/alerts when online (e.g., new lessons, reminders).

Summary by Role:

Role	App Responsibilities
Student	Deliver lessons offline, track progress, offer feedback and motivation
Teacher	Enable lesson uploads, track student interaction, receive admin feedback
Admin	Manage users, content, payments, reports, and system settings
School	Manage groups of users, track school-wide engagement, manage licensing/payment

Key Admin User Responsibilities:

- Manage teacher accounts (approve, suspend, delete)
- Review and approve uploaded lessons before publishing
- Monitor app usage and performance analytics
- Manage student accounts (if needed)
- Manage content categories and curriculum structure
- Handle system settings, backups, and data management

Flow Outline:

Admin Login

- Admin logs into a secure admin dashboard on the app or web portal.
- Teacher Account Management
- Admin reviews new teacher registration requests.
- Approves or rejects teachers to upload lessons.
- Lesson Approval
- Teachers upload lessons (preschool, primary, secondary).
- Admin reviews lessons for quality, appropriateness, and curriculum compliance.
- Approves lessons to be published offline in the app.
- Rejected lessons are sent back to teachers for revision.
- Student Account Oversight
Admin can view student registrations and manage accounts if necessary.

Analytics & Reporting

- Admin monitors app usage stats: number of active students, lessons accessed, success rates.
- Generates reports to improve content and app performance.
- System Maintenance
- Admin manages app settings, performs backups, and oversees data integrity.
- Teacher & Student Support
- Admin can respond to escalated support issues or feedback.
- |-- Maintain System Settings & Backups

Admin Dashboard UI Flow

1. Login Page

- Fields: Username/Email
- Password
- Actions:
- Login
- Forgotten Password
- Two-factor authentication (optional)

2. Dashboard (Home Screen)

- Top Bar:
- Admin profile (with dropdown: Profile, Settings, Logout)
- Notifications (new teacher requests, lesson approvals, system alerts)

Main Panels:

- User Management Summary: Number of active teachers, pending teacher approvals, active students
- Lesson Status Summary: Number of lessons pending approval, approved, rejected
- System Analytics: Daily/monthly active users, lesson downloads, engagement metrics

Quick Actions:

- Approving new teachers
- Review pending lessons
- Generate reports

3. User Management Section

3.1 Teachers Management

List View:

- Search, filter (approved, pending, suspended)
- Columns: Name, Email, Status, Date Joined, Number of Lessons Uploaded, Action buttons (View, Approve, Suspend, Delete)
- Teacher Profile View:
 - Personal info
 - Uploaded lessons summary
 - Activity logs
 - Buttons: Approve/Reject, Suspend/Activate

3.2 Students Management (optional)

Similar list and profile views but with limited actions (View, Suspend, Delete)

4. Lesson Management Section

Pending Lessons Queue:

List of lessons submitted by teachers awaiting approval

- Columns: Lesson Title, Teacher Name, Subject, Level (Preschool, Primary, Secondary), Date Submitted, Preview Button, Approve/Reject buttons

Lesson Review Page:

- Full lesson content preview (text, audio, video, quizzes)
- Comments/feedback box for communicating with teacher
- Action buttons: Approve, Reject (with reason)
- Approved Lessons:
 - Search and filter approved lessons
 - Edit metadata (categories, tags) if needed
- Rejected Lessons: View rejected lessons and reasons

5. Analytics & Reporting Section

- Usage Stats: Active users (daily, weekly, monthly)
- Lessons accessed/downloaded per time frame
- Student engagement metrics (time spent, quiz success rate)

Reports:

- Exportable reports (CSV, PDF)
- Custom date range selection
- Summary charts (bar graphs, pie charts)

6. System Settings

- Manage categories and curriculum structure
- Manage app version releases and offline content sync rules
- Backup and restore data
- Manage notification settings (email, push)

7. Support & Feedback

- View messages and feedback from teachers and students
- Respond or escalate issues
- View system alerts and logs

Teacher Dashboard UI Flow

1. Login Page

- Fields: Email/Username
- Password
- Actions: Login
- Forgotten Password
- Two-factor authentication (optional)

2. Dashboard (Home Screen)

Top Bar: Teacher profile menu (Profile, Settings, Logout)

- Notifications (lesson approval status, feedback from admin)
- Main Panels: My Lessons Summary: Number of lessons uploaded, approved, pending, rejected
- Upload New Lesson: Quick access button
- Recent Feedback: Comments from admin on lessons needing revision
- Usage Stats: Number of students accessing their lessons (if available offline, this may sync when online)

3. Lessons Management Section

3.1 My Lessons List

List View: Search and filter (All, Pending, Approved, Rejected)

Columns: Lesson Title, Subject, Level (Preschool, Primary, Secondary), Date Submitted, Status, Actions (View, Edit, Delete, Resubmit)

3.2 Upload New Lesson

- Upload Form: Title
- Subject :Level (Preschool, Primary, Secondary, Vocational)
- Upload Content (Text, Audio, Video, Quiz) – supports offline-friendly formats
- Description/Notes
- Submit for Approval button

3.3 Lesson Detail / Preview

- View lesson content as students would see it
- View feedback/comments from admin (for rejected or revision requested lessons)
- Edit and resubmit if rejected

4. Notifications

- List of notifications such as:
- Lesson approved
- Lesson rejected with comments
- System updates or announcements

5. Profile & Settings

- Edit personal information
- Change password
- Set preferences (notification preferences, language)

6. Help & Support

- Access FAQs
- Contact support (message admin or support team)

Student Dashboard UI Flow

1. Login / Registration Page

- Fields:
- Username or Student ID
- Password or PIN (optional)

Actions:

- Login
- Register (if allowed)
- Forgot Password / PIN recovery

2. Dashboard (Home Screen)

- Top Bar: Student profile (View/Edit basic info)
- Notifications (new lessons available, messages from teacher/admin)
- Main Panels: My Courses / Levels: List of subjects or levels assigned/enrolled (Preschool, Primary, Secondary)
- Available Lessons: List of lessons approved and available offline for study
- Recently Accessed Lessons: Quick access to last lessons studied
- Progress Summary:
- Lessons completed
- Quiz scores
- Learning streak or engagement badges

3. Lesson Access

- Lessons List: Filter by subject or level
- Search lessons by title or topic
- Lesson Detail Page:
- View lesson content (text, audio, video, quizzes) offline
- Interactive quizzes or exercises integrated
- Option to bookmark or mark lesson as complete

4. Progress & Reports

- View detailed progress on lessons and quizzes
- Summary charts showing strengths and areas to improve
- Downloadable or printable progress reports (if internet connection available)

5. Notifications

- New lessons available
- Feedback or messages from teachers or admin
- App updates or announcements

6. Profile & Settings

- View/Edit basic profile details (name, class/grade, contact info)
- Set preferences (language, notification preferences)

7. Help & Support

- Access FAQs or help guides
- Contact support or teacher (if app supports messaging)

When and how students can make payments to access SomaSmart :Suggest

When Should Students Make Payment?

Option 1: Before First Use (Upfront Payment)

Students (or their parents/guardians) pay once before activating full access to the app. This unlocks all lessons and features for offline use. This method is simple and well-suited for areas with limited internet.

Option 2: After a Free Trial Period

Allow students to try the app free for a limited time (e.g., 7–14 days) or a limited number of lessons. After the trial, payment is required to continue accessing content.

Option 3: Subscription or Periodic Payment

Access is granted for a fixed duration (monthly, quarterly, or yearly). Students renew their subscription to maintain access. This is ideal if content is frequently updated.

Option 4: Pay Per Lesson or Module

Students pay for individual lessons or modules. This allows flexibility but requires a system to manage access per lesson offline.

How Should Students Make Payment?

1. Mobile Money Payments (e.g., M-Pesa, TigoPesa, Airtel Money)

Highly accessible and popular in Tanzania. Students' complete payments via USSD codes or mobile money apps. Payment confirmation activates the app or credits the student's account.

2. In-App Payment Gateway

If internet is available at least initially, integrate local gateways (Flutter wave, Paystack) to accept cards, mobile money, or bank payments. This generates an activation code or unlocks content immediately.

3. Voucher or Coupon Codes

Prepaid coupon codes or vouchers can be purchased from agents, schools, or shops and entered the app to unlock content offline. This allows students without mobile money or internet access to lessons easily. Coupons can also be distributed **as scholarships or promotional offers.**

4. Agent or Vendor Payment

Students pay cash to authorized agents or kiosks, who then provide activation codes or coupons to unlock the app's content.

5. School or NGO Sponsorship

Schools or NGOs pay in bulk for student access licenses, allowing students to use the app for free or at a discounted rate.

Recommended Payment Flow Example

Student downloads the app and accesses limited content or trial lessons.

When full access is needed, the app prompts the student to make payment.
Students select a payment method:

Mobile Money: Follows payment instructions and completes transaction; app verifies payment and unlocks content.

Coupon Code: Student enters the coupon code received from an agent, school, or promotion; app unlocks content offline immediately.

- In-App Payment: Student completes payment via integrated gateway; app unlocks content.
- Upon successful payment or coupon redemption, the app grants full offline access to lessons.

Additional Recommendations

- Keep payment instructions clear and simple within the app.
- Support multiple payment methods to suit different user preferences.
- Use SMS or app notifications to confirm successful payments or coupon redemptions.
- Provide customer support to assist with payment or activation issues.
- Offer discounted or free coupons for low-income students or special promotions.

Coupon Code Management Workflow

1. Coupon Creation

- Admin Dashboard: Create Coupon
- Admin logs into the Admin Dashboard.
- Navigates to Coupon Management section.
- Clicks Create New Coupon.
- Enters coupon details:
 - Coupon Code (auto generated or custom)
 - Discount Type (e.g., full access, partial access, % discount)
 - Validity Period (start and expiry dates)
 - Usage Limits (e.g., number of redemptions per coupon or per user)
 - Target Audience (general, specific schools, groups, or individual users)
 - Notes or description (optional)
- Saves and activates the coupon.
- System generates unique coupon codes (if bulk creation).

2. Coupon Distribution

- Methods of Distribution:
 - Printed scratch cards or vouchers sold/distributed by agents or schools.
 - Digital coupons sent via SMS, email, or messaging apps (WhatsApp, Telegram).
 - Promo codes shared in events or social media campaigns.
 - Bulk coupon codes allocated to NGOs or schools for distribution to beneficiaries.
- Tracking:
 - Admin dashboard tracks which coupons were distributed, to whom, and when.

3. Coupon Redemption

- Student App:
- Student opens the app and goes to the Payment/Unlock Content section.
- Selects Redeem Coupon option.
- Enter the coupon code manually or scan the QR code (if applicable).
- App validates coupon:
- Check if coupon exists and is active.
- Check if coupon is within validity period.
- Checks usage limits (per user and overall).
- If valid, the app unlocks the corresponding content/features offline immediately.
- If invalid or expired, the app shows an appropriate error message.
- Optional: Student can view redeemed coupons and remaining validity/access period.

4. Coupon Usage Tracking & Reporting

- Admin Dashboard:
- View reports on coupon redemptions:
- Each coupon was redeemed.
- User details (anonymized if needed).
- Date/time of redemption.
- Active vs expired coupons.
- Export reports for accounting or auditing.
- Alerts: Notifications about if a coupon is near usage limit or about to expire.

5. Coupon Expiry & Deactivation

- Coupons automatically become inactive after expiry date or after max usage reached.
- Admin can manually deactivate coupons if needed (fraud prevention, errors).
- Deactivated coupons cannot be redeemed.

Payment management workflow

Student → Choose Payment Option

- └─ Mobile Money → Pay → Verify → Unlock
- └─ Coupon Code → Enter → Validate → Unlock
- └─ Voucher Code → Enter → Validate → Unlock
- └─ Sponsored → School/NGO checks → Unlock

Payment Success → Access Granted → Confirmation Sent

↓

Logs stored locally → Synced to Admin

↓

Admin Dashboard → Track, Report, Manage

1. Payment Initiation (Student Side)

Trigger: Student completes trial or attempts to access locked content.

Student Action: Navigates to “Unlock Full Access” or “Make Payment” section.

Payment Options Displayed:

- Mobile Money (M-Pesa, TigoPesa, Airtel Money)
- Coupon Code
- Voucher Code / Scratch Card
- School or NGO Sponsored Access (if applicable)

2. Payment Methods Workflow

2.1 Mobile Money Payment

- Student selects Mobile Money → Chooses provider.
- The app displays USSD code or payment instructions.
- Student completes payment outside the app (e.g., via SIM toolkit or mobile money app).
- App verifies payment: Online: API confirmation with payment gateway.
- Offline: Student enters transaction ID, or app waits for periodic sync.
- On success: Access unlocked + payment receipt stored locally & synced when online.

2.2 Coupon Code Redemption

- Student selects Redeem Coupon.
- Enters or scans code.
- App validates:
 - Code exists and is active
 - Not expired or overused
- On success: App unlocks access, logs redemption locally and syncs later if offline.

2.3 Voucher / Scratch Card

- Students enter voucher code (from physical card).
- Follows the same process as coupon code.
- Useful in areas with no internet/mobile money access.

2.4 Sponsored Access (School/NGO)

- App checks if student is linked to a prepaid license via school or NGO.
- If yes, apps unlock content without additional payment.
- Ideal for bulk sponsorships or funded deployments.
- 3. Payment Validation & Access Control
 - Online:
 - Real-time API verification with payment provider or coupon DB.
 - Confirmation triggers full content access.
 - Offline:
 - Local validation of preloaded coupon/voucher code rules.
 - Temporary unlocks synced to server later when internet is available.

4. Payment Confirmation & Receipt

- Student receives:
- Success message
- Receipt/confirmation code (stored in app)
- SMS or push notification if enabled
- Access duration or license type shown (e.g., "Access until 30 Sept 2025")

5. Admin Portal – Payment Oversight

- Admin Can: Create and distribute coupon/voucher codes
- Track all payments (mobile money, codes)
- View redemption logs per user, region, or school
- Revoke or extend access
- Generate financial reports and analytics

6. Data Sync and Audit Trail

- All payment actions are logged locally.
- When online:
- The app syncs payments, activations, and coupon redemptions to server.
- Admin dashboard reflects real-time updates.
- Logs maintained for auditing.

How to Include Schools in SomaSmart

✓ 1. Create a Dedicated “School Account” System

◆ School Admin Role:

- Schools sign up and receive a School Admin Dashboard.
- Can register and manage their students and teachers.
- Can monitor student progress and lesson usage (syncs when online).
- Can you upload or recommend lessons (optional).
- Can view learning analytics (quiz results, engagement).

✓ 2. Offer School Licensing Packages

- Payment Model Options:
- Per-school license (unlimited students up to X users)
- Per-student license with discounted bulk pricing
- Sponsored model via NGO/CSR partner pays for the school
- Schools can purchase access codes or receive sponsored ones to distribute.

✓ 3. Student Enrollment by Schools

Two Options:

- Bulk Registration:
- School admin uploads a list of students (name, grade, unique ID).
- System generates access codes for each student.
- Student Self-Registration:
- Students select their school from a dropdown.
- Enter school ID or teacher-issued code for authentication.

✓ 4. Teacher Integration at School Level

- Teachers can:
- Upload lessons (linked to their school)
- Assign specific lessons to their class
- Review student activity if data syncs
- Optional: Teacher chat/support group to collaborate with app team

✓ 5. Offline Distribution to Schools

- Schools receive the app installer + offline content via:
- Preloaded memory cards/USB drives
- School tablets or computers
- Local partner hubs or education officers

✓ 6. School Dashboard Features

- School Profile (location, type, contact info)
- User Management (teachers, students)
- Payment & License Management
- Usage Analytics:
- % of students active
- Top lessons accessed
- Quiz pass rates

✓ 7. Support and Training

- Offer onboarding and training to school heads and ICT teachers.
- Provide physical or downloadable guides.
- Host webinars or WhatsApp groups for continuous support.