

Product Requirement Document (PRD)

Project Title: StrawPoll Replica

Product Manager: Teja

Product Name: Client has not decided yet

Date: Jan 2026

Version: 1.0

1) Objective & Vision

Build a lightweight, mobile app-based real-time polling platform that allows users to create, share, vote, and analyze polls with optional meeting scheduling tools — replicating the key functionality of StrawPoll.com for broad use cases (social, business, education).

2) User Personas

A. Guest User

- Create simple polls without an account.
- Vote in polls.

B. Registered User

- Manage all polls created.
- Export results.
- Use advanced settings and integrations.

C. Admin

- Maintain site content & security.
- Moderate reported polls.
- Admin can add/delete votes

D. API Consumer

- Create/manage polls via API integration.
-

3) User Stories

Poll Creation

- *As a user, I want to create a poll with a question and options, so others can vote.*
- *As a user, I want to share my poll via URL or embed code.*

Voting

- *As a participant, I want to vote on a poll and see live results.*
- *As a poll creator, I want to limit voting (unique votes per IP/session/code).*

Customization

- *As a creator, I want to add emojis, set deadlines, and manage visibility.*

Analytics & Export

- *As a registered user, I want export results for offline reporting.*

Meeting Scheduling

- *As a user, I want to create polls to schedule group meetings.*

Administration

- *As an admin, I want to block spam and manage content reports.*

4) Functional Requirements

Feature	Description	Priority
Poll creation UI	Create question, options, deadlines, settings	★★★★★
Voting engine	Count votes, enforce duplicates rules	★★★★★
Real-time result charts	Display live responses with charts	★★★★★
User accounts	Sign up/login with email	★★★★
Results export	CSV/Excel export	★★★★
Poll API	REST endpoints for poll CRUD	★★★★
Meeting scheduler	Create/publish date/time polls	★★★★
Custom link & branding	Premium feature	★★★
Ads system	For free tier	★★★
CAPTCHA & bot checks	Premium or optional	★★★

5) Non-Functional Requirements

- **Scalability:** Handle large numbers of votes in real time.
- **Security:** Protection against spam bots, fraud, and injection attacks.
- **Performance:** Sub-second response for vote casting.
- **Accessibility:** Mobile-friendly, responsive UI.

- **Compliance:** GDPR and privacy standards.
-

6) UI/UX Flow

1. **Landing Page:** CTA to Create Poll, Schedule Meeting, Login.
 2. **Poll Builder:** Inputs for question, options, settings.
 3. **Poll View Page:** Voting options + real-time results.
 4. **Dashboard:** Manage polls, export results.
 5. **Admin Panel:** Reports, moderation tools.
-
-

8) API Endpoints

- POST /polls – Create poll
 - GET /polls/:id – Get poll
 - POST /polls/:id/vote – Cast vote
 - GET /polls/:id/results – Get results
 - GET /user/polls – Get user's polls
-

9) Milestones & Timeline

Phase	Duration
Discovery & Design	2 weeks
MVP Development	6 weeks
Testing & QA	2 weeks
Beta Release	1 week
Public Launch	1 week

10) Success Metrics

- Daily active polls created
- Poll participation rate
- System uptime
- API adoption

- Conversion from free to paid plans