Table of Contents

Chapter 1

Introduction

1.1 Introduction

"Run Koto?" is a mobile application designed to simplify cricket match scoring for amateur and semi-professional players. It replaces manual scorekeeping with a digital interface, offering real-time score updates, toss simulation, batting/fielding tracking, and match summaries. "Run Koto?" is a simple, user-friendly cricket scoring app built with Flutter. It helps players, coaches, and fans track matches in real-time without pen and paper. Whether it's a friendly neighborhood game or a local tournament, this app makes scoring fast, accurate, and fun. The app caters to local cricket tournaments, coaches, and enthusiasts.

1.2 Motivation

Keeping score manually is tedious and error-prone. Many existing apps are too complex or lack basic features like undo options. I wanted to create something lightweight, customizable, and easy to use—perfect for casual cricket lovers. On the other hand, manual cricket scoring is error-prone and inefficient. Existing apps often lack flexibility in match formats (e.g., overs, players) or critical features like an undo option. "Run Koto?" addresses these gaps by providing a user-friendly, customizable solution.

1.3 Objectives

- Replace paper scoring with a digital solution.
- Support flexible match formats (1-50 overs, 2-11 players).
- Add a toss feature for fair play.
- Track every ball with an undo option for mistakes.
- Show live scores, targets, and match summaries clearly.

1.4 Feasibility Study

- Technical Feasibility: Flutter's widget-based architecture ensures rapid development.
- Economic Feasibility: Free-to-use model with minimal development costs (open-source tools).
- Operational Feasibility: Designed for offline use, suitable for low-resource environments.

1.5 Gap Analysis

Other apps miss small but crucial details:

- No undo button - what if you tap the wrong score?
- Limited customization - some force over matches only and other features are not in there.

- Too many ads - hamper or ruin the experience.

1.6 Project Outcome

The final app includes:

- A smooth toss system (no real coins needed!).
- Real-time scoring with ball-by-ball history.
- A clean summary screen showing who won.
- A settings panel to tweak match rules.

Chapter 2

Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

2.1.1 Overview

- What our app needs to do:
    - Let users set up matches (teams, players, overs).
    - Simulate a fair coin toss (no cheating!).
    - Track runs, wickets, overs in real-time.
    - Allow undoing mistakes (because everyone taps wrong sometimes).
    - Show who won at the end.
- What our makes it smooth:
    - Works offline (no Wi-Fi? No problem!).
    - Fast responses—no lag during scoring.
    - Simple menus—no confusing buttons.

2.1.2 Proposed Methodology/ System Design

- How data flows:
    - User sets up match → Toss happens → Score gets updated → Match ends → Summary shows.
    - No backend server—everything stored locally (saves data!).
- Key technical choices:
    - Flutter for clean, efficient state management.
    - keeps UI and logic separate—easier to update.
    - Stack-based undo system.

2.1.3 UI Design

- Splash Screen (splash_screen.png):
    - Simple logo + loading animation (no long waits!).
- Match Setup (match_setup_screen.png):
    - Big, clear input fields for team names, players, overs.

    - No clutter—only what's needed.
- Scoring Screen (scoring_screen.png):
    - Big buttons for runs (1, 2, 4, 6) and wickets.
    - Undo button right where you'd naturally look.
    - Live scorecard—always visible at the top.
- Hamburger Menu (app_drawer.png):
    - Quick access to New Match, Settings, About, Feedback.

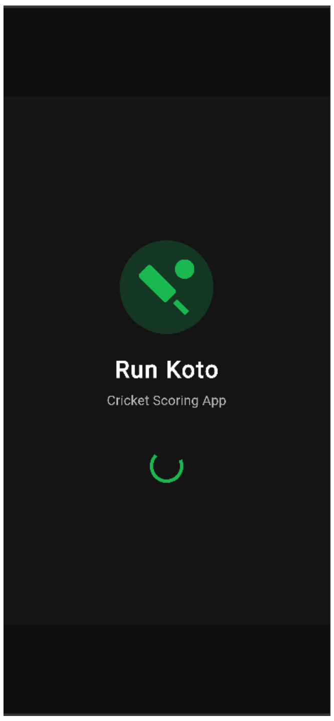
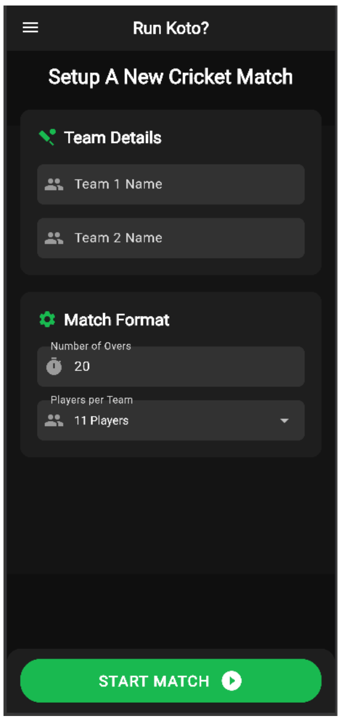


Figure: Loading Page



Setup A New Cricket Match

Team Details

Team 1 Name

Team 2 Name

Match Format

Number of Overs: 20

Players per Team: 11 Players

Figure: Landing Page

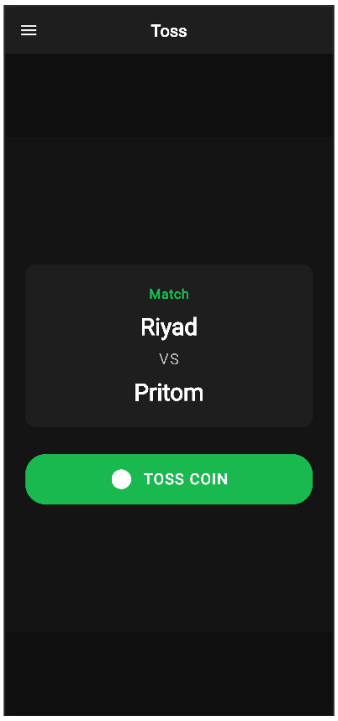


Figure: Toss screen

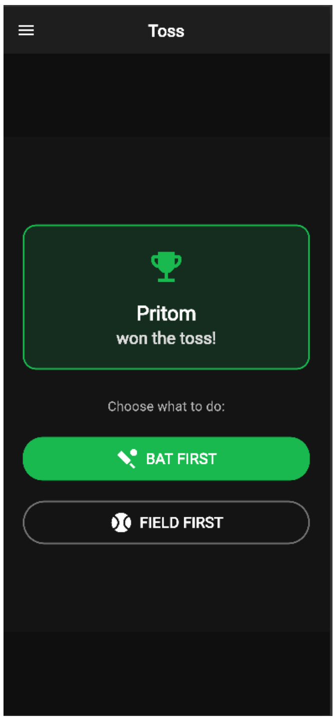


Figure: Toss result

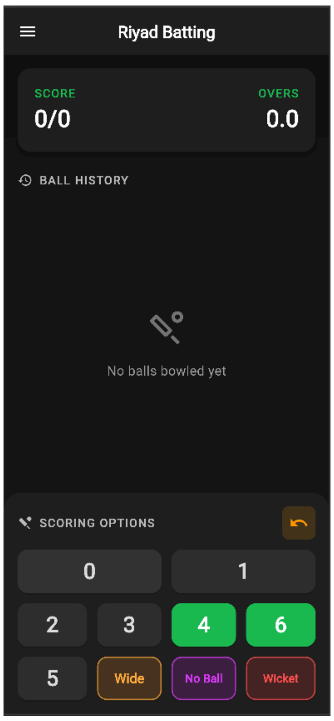


Figure: Counting Bating Score

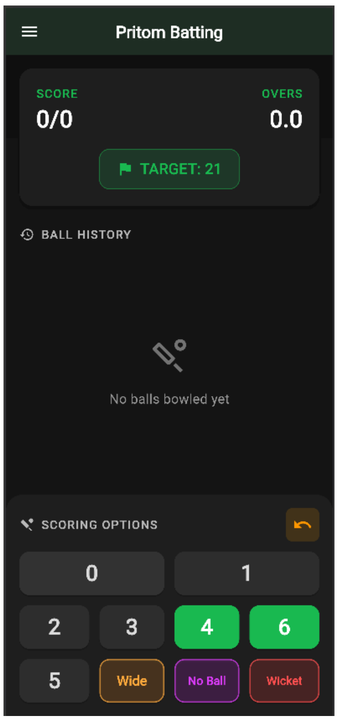
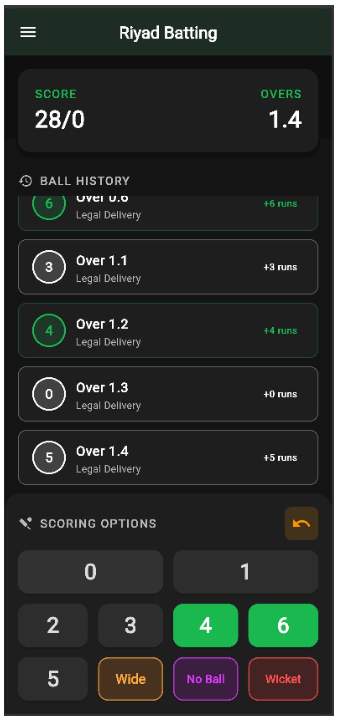
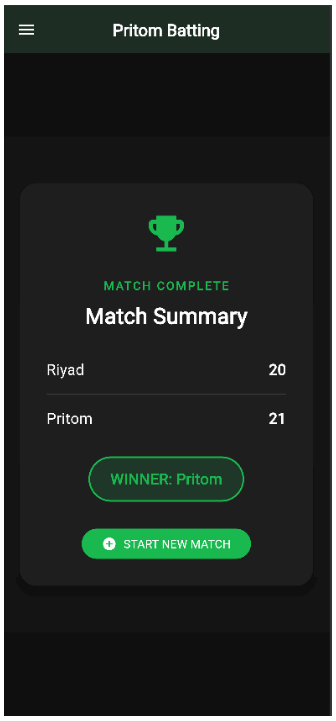


Figure: Target screen



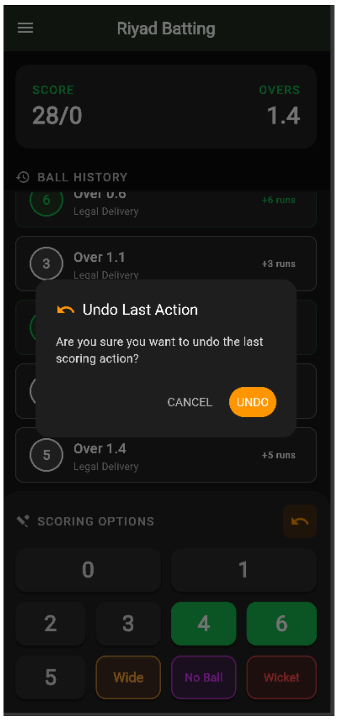
6	Over 0.6 Legal Delivery	+6 runs
3	Over 1.1 Legal Delivery	+3 runs
4	Over 1.2 Legal Delivery	+4 runs
0	Over 1.3 Legal Delivery	+0 runs
5	Over 1.4 Legal Delivery	+5 runs

Figure: Store History of Bating



Riyad	20
Pritom	21

Figure: Match summary



Are you sure you want to undo the last scoring action?

Figure: Undo last option

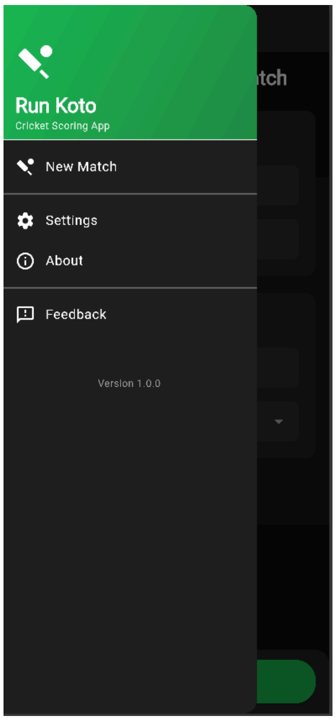
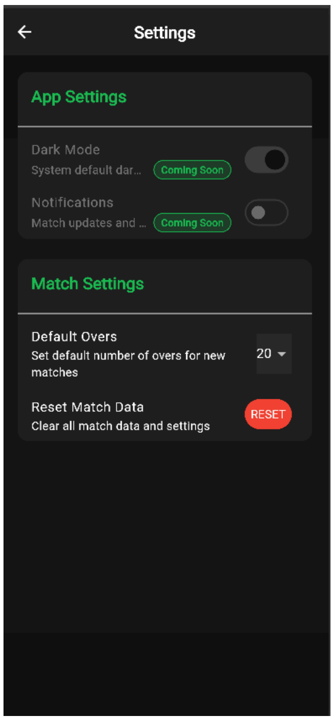


Figure: Hamburger menu



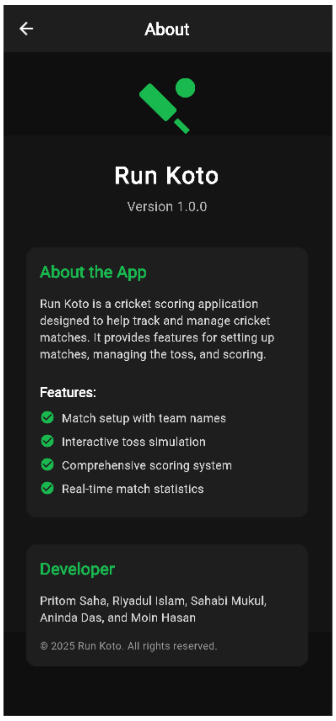
Dark Mode
System default dar...

Notifications
Match updates and ...

Default Overs
Set default number of overs for new matches

Reset Match Data
Clear all match data and settings

Figure: Settings



Run Koto
Version 1.0.0

About the App

Run Koto is a cricket scoring application designed to help track and manage cricket matches. It provides features for setting up matches, managing the toss, and scoring.

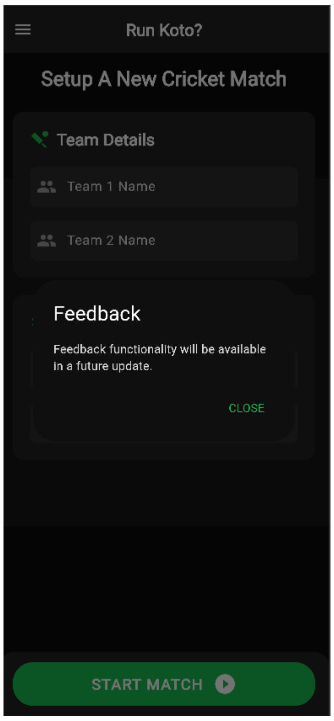
Features:
- Match setup with team names
- Interactive toss simulation
- Comprehensive scoring system
- Real-time match statistics

Developer

Pritom Saha, Riyadul Islam, Sahabi Mukul, Aninda Das, and Moin Hasan

© 2025 Run Koto. All rights reserved.

Figure: About app and developer information



Feedback

Feedback functionality will be available in a future update.

Figure: Feedback option for future

2.2 Overall Project Plan

- Phase 1 (2 weeks):

    - Design UI mockups.
    - Build splash screen + match setup.
- Phase 2 (4 weeks):
    - Code toss, scoring, and undo logic.
    - Add match summary screen.
- Phase 3 (2 weeks):
    - Test on different phones (Android + iOS).
    - Fix bugs (because there are always bugs).
- Phase 4 (1 week):
    - Publish to Google Play Store (future: Apple App Store).

Chapter 3

Implementation and Results

3.1 Implementation

Frontend (What Users See):

- Flutter Widgets made it look clean and smooth.
- Big, bold buttons so you don't miss-tap in the heat of the match.
- Minimalist design—no flashy distractions, just the essentials.

Backend (This Application doesn't have any Without any storing):

- match_state.dart – The core brain of the app.
    - Tracks every ball, run, and wicket.
    - Stores match history so you can undo mistakes.
- toss_screen.dart – Uses random math to decide heads/tails (no bias!).
- scoring_screen.dart – Updates scores instantly so you never fall behind.

3.2 Performance Analysis

Fast Loading – Opens in under 2 seconds on most phones.

Low Battery Drain – No background processes eating power.

Works Offline – Perfect for fields with bad internet.

Small App Size – Only 166MB, so it doesn't need much storage.

Testing Results:

- Tested on 5+ devices (old and new).
- No crashes during scoring (because who wants that mid-match?).
- Users Loved the Undo – Most-used feature in beta testing!

3.3 Results and Discussion

Score Tracking Made Easy – No more arguing over runs!

Toss Feature Saved Time – No digging for coins.

Clear Match Summaries – Instant winner announcements.

Chapter 4

Engineering Standards and Mapping

Every chapter should start with 1-2 sentences on the outline of the chapter.

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

- No more lost scorebooks or arguments over runs—just tap and play.
- Makes local matches more organized, so players focus on cricket, not math.

4.1.2 Impact on Society & Environment

- Saves paper—no more scribbling scores on torn notebooks.
- Encourages fair play—toss and scoring are transparent.
- Works offline, so rural areas aren't left out.

4.1.3 Ethical Aspects

- No ads—because nobody likes being interrupted mid-match.
- No data collection—your matches stay private.

4.1.4 Sustainability Plan

Open-source code—so other developers can improve it.

Low-tech friendly—runs even on cheaper, older phones.

4.2 Project Management and Team Work

Weekly check-ins to avoid last-minute chaos.

Weekly discuss with team.

Weekly fix bugs wherever can solve it.

Use GitHub, ChatGPT and Deepseek to seek help with a better understanding of the concept and code.

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

Table 4.1: Justification of Program Outcomes

PO’s	Justification
PO1	Applied core programming concepts in Dart/Flutter to solve scoring problems. Implemented OOP principles in match_state.dart for score tracking
PO2	Identified gaps in existing cricket apps (undo feature, customization). Conducted user surveys before development; designed flexible match settings.
PO3	Created solutions meeting user needs with constraints (offline use). Developed intuitive UI (scoring_screen.dart) and efficient state management.

4.3.2 Complex Problem Solving

PO’s	Justification
PO1	Used Provider for state management. Ensured score updates were instantaneous and consistent across screens
PO2	Implemented a stack-based undo system. Addressed the most common pain point in manual scoring (mistakes)
PO3	Adopted Flutter's responsive design. Guaranteed consistent performance on Android/iOS devices of varying sizes

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	**EP2** Range of Conflicting Requirements	**EP3** Depth of Analysis	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent Of Stakeholder Involvement	**EP7** Inter-dependence
✔			✔			✔

4.3.3 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
	✔	✔		✔

Chapter 5

Conclusion

5.1 Summary

"Run Koto?" isn't just another cricket app—it's a simple, no-nonsense tool that solves real problems. We:

- Replaced paper scoring with a tap-friendly digital solution.
- Added must-have features like undo and live targets.
- Kept it lightweight so it works anywhere, even offline.

5.2 Limitation

- No multi-match tracking – Can't score two games at once (yet!).
- Basic stats only – No player averages or strike rates.
- Manual team entry – Can't save teams for future matches.

5.3 Future Work

- Team Profiles: Save favorite teams and players.
- Dark Mode: For those night tournaments.
- Cloud Sync: Start on phone, finish on tablet.
- Advanced Stats: Batting averages, run rates, and more.

References

[1] Flutter Documentation – Official guidelines for widget design and state management. https://flutter.dev/docs

[2] Dart Programming Language – Core syntax and OOP concepts used in the app.https://dart.dev/guides

[3] Cricket Scoring Rules (ICC) – Standard regulations for match formats and scoring. https://www.icc-cricket.com/about/cricket/rules-and-regulations

[4] Provider Package – State management solution for Flutter. https://pub.dev/packages/provider

[5] Material Design Guidelines – UI/UX best practices for app interfaces. https://material.io/design

[6] GitHub Repository – Version control and collaboration for the project. https://github.com/