

PRODUCT REQUIREMENTS DOCUMENT (PRD)

CollabX – Firebase-Based Student Collaboration Platform

1. Problem Statement

Students often struggle to find suitable project partners with compatible skills, availability, and commitment levels. Existing methods like WhatsApp groups, classroom announcements, or random pairing lead to mismatches, spam, privacy issues, and incomplete projects.

There is no centralized, structured, and privacy-first platform that enables students to discover and connect with potential collaborators efficiently and securely.

1. Objective

To build a web-based collaboration platform that allows students to:

Create structured collaboration profiles

Discover compatible project partners

Send and manage collaboration requests

Reveal contact details only after mutual acceptance

Ensure privacy, security, and minimal misuse

The platform prioritizes intent, privacy, and simplicity, making it ideal for hackathons and academic use cases.

1. Target Users & Roles

Primary Users

College students

Hackathon participants

Mini-project / Final-year project students

User Roles

Student (Primary Role)

Sign up and log in securely

Create and update profile

Discover other students

Send collaboration requests

Accept or reject requests

View matches and contact details after acceptance

Admin (Moderation Role – Limited Scope)

Log in securely

View reported users

Take moderation actions (warn / suspend / dismiss)

Ensure platform safety and trust

1. Proposed Solution

CollabX is a Firebase-powered collaboration platform that provides:

Secure authentication using Firebase Authentication

Structured user profiles stored in Firestore

Partner discovery with explicit collaboration intent

Request-based connection flow (accept / reject)

Contact detail visibility only after mutual match

Admin moderation using role-based access control

Firebase eliminates backend complexity while ensuring scalability and security.

1. User Flow

User lands on the website

User signs up or logs in (Firebase Auth)

User completes profile creation

User navigates to Discover Partners

User browses partner cards

User sends collaboration request

Recipient accepts or rejects the request

Dashboard updates request status

On acceptance, contact details are revealed

1. Screens & Pages

1 Landing Page

Product overview

Key benefits

Call-to-action: Find Project Partners

2 Authentication Pages

Sign Up (Name, Email, Password)

Log In (Email, Password)

3 Profile Page

Name

Short bio

Skills (tags)

Availability

Project links

Contact details (hidden until match)

4 Discover Partners Page

Partner cards displaying:

Name

Skills

Availability

Projects

"Connect" button

Disabled state if request already sent

5 Dashboard Page

Received Requests (Accept / Reject)

Sent Requests (Pending / Accepted / Rejected)

Matches with contact details

6 Admin Portal

Admin login

View reported users

Resolve reports

Moderate misuse

1.  Functional Requirements (MVP)

Authentication

Email & Password authentication

Session persistence

Secure logout

Auth state protection for private pages

Profile Management

Create and update user profile

Profile data stored in Firestore

Skills, availability, projects editable

Contact details hidden by default

Partner Discovery

Fetch all user profiles except current user

Display structured partner cards

Prevent duplicate collaboration requests

Collaboration Requests

Send collaboration requests

Accept or reject requests

Track request status in real time

Dashboard

View received requests

View sent requests with status

Accepted requests shown as matches

Match & Contact

Reveal mobile number and LinkedIn only after acceptance

Enforced via Firestore Security Rules

Admin & Moderation

Admin-only access using Firebase Custom Claims

View and resolve reports

Maintain platform safety

1.  Safety, Privacy & Misuse Prevention

Key Risks

Spam requests

Fake profiles

Harassment after contact exchange

Mitigation

No direct contact before acceptance

Contact details hidden by default

Request-based interaction model

Admin moderation

Firebase Security Rules enforcement

1.  Non-Functional Requirements

Responsive UI (desktop-first)

- Fast page loads
- Secure authentication
- Minimal backend complexity
- Scalable infrastructure via Firebase

1. Out of Scope

- In-app chat
- AI-based matching
- Group/team creation
- Notifications (email/push)

- Resume uploads
- Automated moderation
- Analytics dashboard

1. Tech Stack

- Frontend

- HTML

- CSS

- JavaScript

- Backend (Firebase)

- Firebase Authentication

- Cloud Firestore

- Firebase Security Rules

- Firebase Custom Claims (Admin)

- Tools

- Firebase Console

- GitHub

- VS Code

- Postman (optional testing)

1. Database Design (Firestore)

- Users Collection

```
users/{uid} {
  name: String,
  email: String,
  bio: String,
  skills: [String],
  availability: String,
  projects: [String],
  contact: {
    mobile: String,
    linkedin: String
  },
  isSuspended: Boolean,
  warningCount: Number,
  createdAt: Timestamp
}
```

- Requests Collection

```
requests/{requestId} {
  fromUserId: String,
  toUserId: String,
```

```
status: "pending" | "accepted" | "rejected",
createdAt: Timestamp
}

Reports Collection
reports/{reportId} {
reportedUserId: String,
reportedByUserId: String,
reason: String,
status: "pending" | "resolved",
actionTaken: "warned" | "suspended" | "dismissed",
createdAt: Timestamp
}
```

1. Security Rules

Users can edit only their own profile

Requests accessible only to sender and receiver

Contact details visible only after match

Admin-only access for moderation

Enforced via Firestore Security Rules

1. Success Criteria (Hackathon)

Users can sign up and log in

Profiles load and save correctly

Partner discovery works

Collaboration requests function end-to-end

Dashboard reflects correct status

Privacy rules are enforced

Admin moderation works

Demo flow is smooth

1. Future Scope

AI-based partner matching

Team formation (multi-user)

In-app chat

Skill verification

College-wide collaboration

Hackathon-specific filters

1. Final Notes

CollabX prioritizes:

Intent over noise

Privacy over exposure

Security over complexity

Simplicity over overengineering

Firebase enables rapid, secure development while aligning perfectly with hackathon timelines and real-world scalability.