Design Document

Data to Track

1. **Events**
   - Event ID, Name, Type (e.g., Hackathon, Workshop)
   - Details, Date, Time, Location, Organizer
2. **Students**
   - Student ID, Name, Email, Password (hashed)
   - Contact Information
3. **Registrations**
   - Registration ID, Event ID, Student ID, Registration Date, Status
4. **Attendance**
   - Attendance ID, Registration ID, Check-in Timestamp, Status (Present/Absent)
5. **Feedback**
   - Feedback ID, Registration ID, Rating (1-5), Comment, Timestamp

Schema

Events

Column	Type	Null?	Default	Constraints / Notes
id	INTEGER	NO	—	PK, AUTOINCREMENT
name	TEXT	NO	—	
type	TEXT	NO	—	(e.g., Hackathon, Workshop)
date	TEXT	NO	—	ISO date string
details	TEXT	YES	NULL	Optional event details

- Primary Key: id
- Foreign Keys: none
- Indexes: UNIQUE(name)
- Example: { id: 1, name: "rohith", password: "****" }

Registrations

Column	Type	Null?	Default	Constraints / Notes
id	INTEGER	NO	—	PK, AUTOINCREMENT
event_id	INTEGER	NO	—	FK → events(id)
student_id	INTEGER	NO	—	FK → students(id)

- Primary Key: id
- Foreign Keys: event_id, student_id
- Indexes: UNIQUE(event_id, student_id)
- Example: { id: 7, event_id: 1, student_id: 2 }

Attendance

Column	Type	Null?	Default	Constraints / Notes
registration_id	INTEGER	NO	—	PK, FK → registrations(id)
present	BOOLEAN	NO	0	1 = present, 0 = not present

- Primary Key: registration_id
- Foreign Keys: registration_id
- Indexes: —
- Example: { registration_id: 7, present: 1 }

Feedback

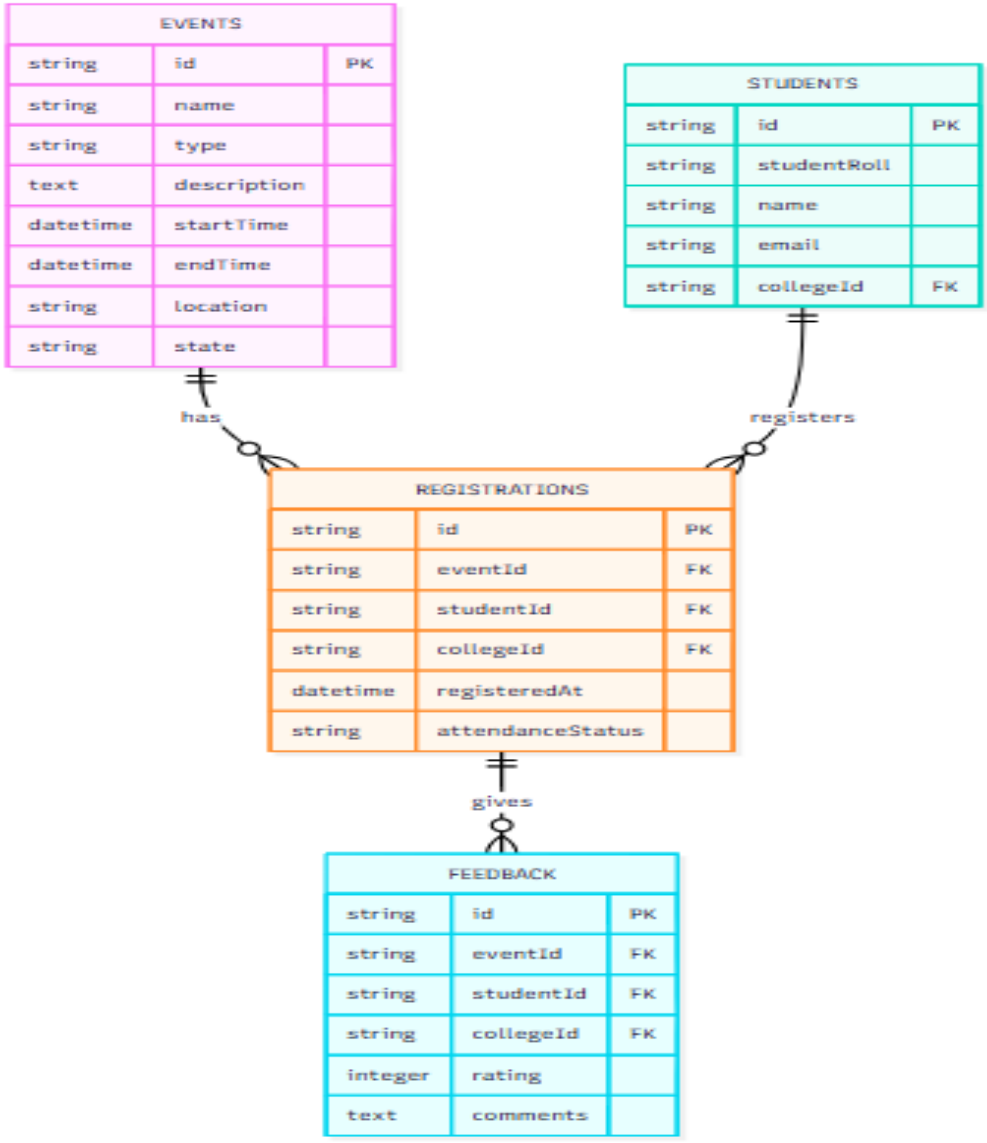
Column	Type	Null?	Default	Constraints / Notes
registration_id	INTEGER	NO	—	PK, FK → registrations(id)
rating	INTEGER	YES	NULL	CHECK (rating BETWEEN 1 AND 5), nullable
comments	TEXT	YES	NULL	Optional

- Primary Key: registration_id
- Foreign Keys: registration_id
- Indexes: —
- Example: { registration_id: 7, rating: 5, comments: "Amazing!" }

PROTOTYPE CODE

```
-- EVENTS TABLE
 CREATE TABLE IF NOT EXISTS events (
   id INTEGER PRIMARY KEY AUTOINCREMENT,
   name TEXT NOT NULL,
   type TEXT NOT NULL,
   date TEXT NOT NULL,
   details TEXT
 );

 -- STUDENTS TABLE
 CREATE TABLE IF NOT EXISTS students (
   id INTEGER PRIMARY KEY AUTOINCREMENT,
   name TEXT NOT NULL UNIQUE,
   password TEXT NOT NULL
 );

 -- REGISTRATIONS TABLE
 CREATE TABLE IF NOT EXISTS registrations (
   id INTEGER PRIMARY KEY AUTOINCREMENT,
   event_id INTEGER NOT NULL,
   student_id INTEGER NOT NULL,
   UNIQUE(event_id, student_id),
   FOREIGN KEY(event_id) REFERENCES events(id),
   FOREIGN KEY(student_id) REFERENCES students(id)
 );
```

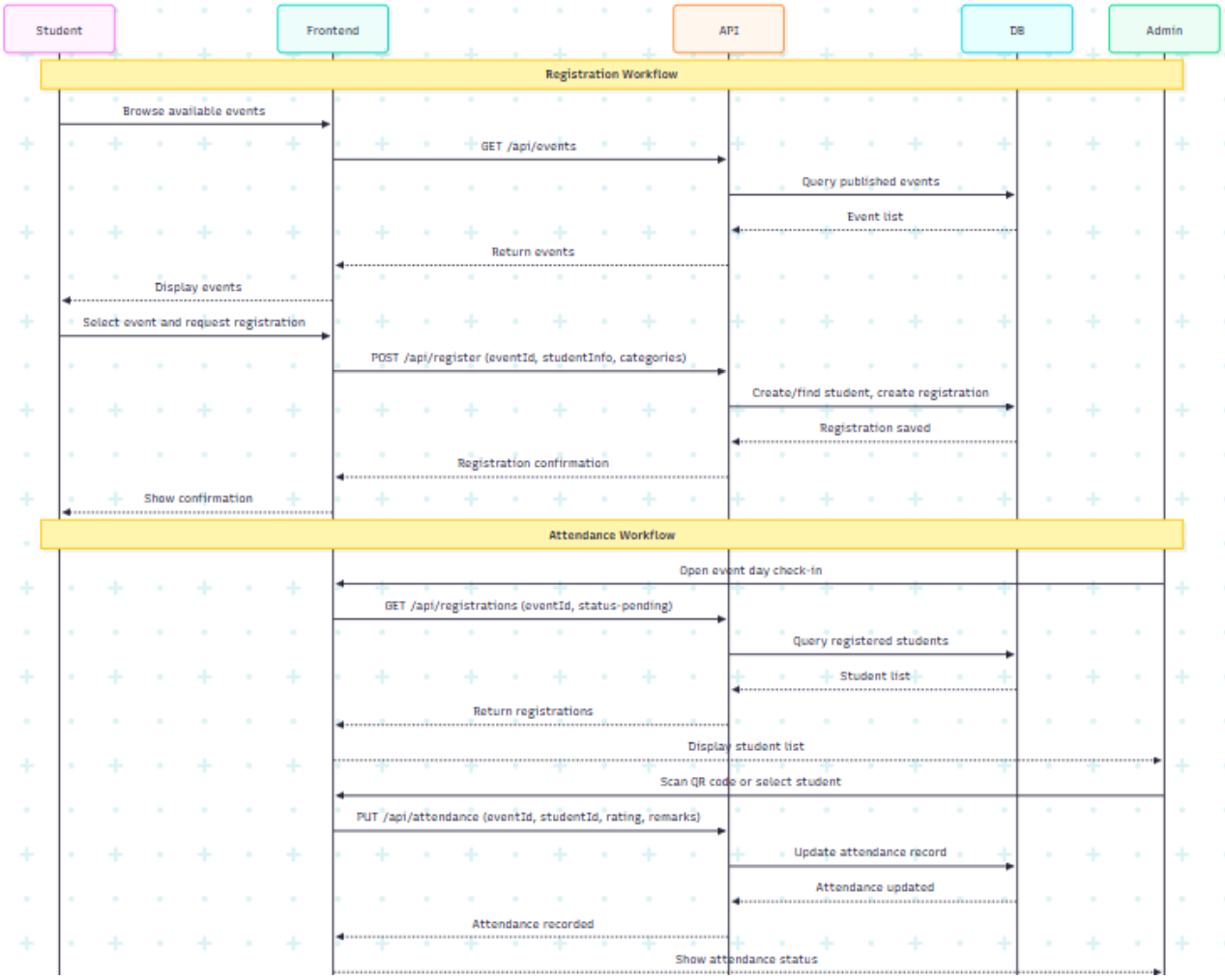
```
-- ATTENDANCE TABLE
CREATE TABLE IF NOT EXISTS attendance (
  registration_id INTEGER PRIMARY KEY,
  present BOOLEAN DEFAULT 0,
  FOREIGN KEY(registration_id) REFERENCES registrations(id)
);

-- FEEDBACK TABLE
CREATE TABLE IF NOT EXISTS feedback (
  registration_id INTEGER PRIMARY KEY,
  rating INTEGER CHECK (rating BETWEEN 1 AND 5),
  comments TEXT,
  FOREIGN KEY(registration_id) REFERENCES registrations(id)
);
```



API Design

Endpoint	Method	Description	Request Body	Response
/event	GET	Get list of events	-	List of events
/events	POST	Create a new event (admin only)	Event details	Created event object
/registrations	GET	Get registrations by student/event	Query: student_id or event_id	List of registrations
/register	POST	Register a student for an event	{ student_id, event_id }	Registration confirmation
/attendance	POST	Mark attendance	{ registration_id, time, status }	Attendance confirmation
/feedback	POST	Submit feedback	{ registration_id, rating, comment}	Confirmation
/reports/registrations	GET	Reporting on registrations	Query parameters (event, date range)	Stats report
/reports/attendance	GET	Attendance reports	Query parameters	Stats report
/reports/feedback	GET	Feedback summaries	Query params	Aggregated feedback

Workflow



Assumptions & Edge Cases

- Duplicate registrations prevented.
- Feedback optional; system handles missing feedback gracefully.
- Cancelled events marked and hidden or flagged.
- Attendance must be done within event time frame.
- Data privacy ensured: passwords stored securely.
- System designed to scale (pagination, caching).
- Notifications and reminders are out-of-scope for MVP.