This one-page overview shows you Firestore schema, data-writing behavior, and auth setup in a clear, concise format.

1. Introduction to Firebase

Firebase is Google's mobile/back-end platform offering services like Authentication, Cloud Firestore (a real-time NoSQL database), Hosting, and Functions. It lets you focus on UI and business logic while Google handles scaling and security.

2. Cloud Firestore

Cloud Firestore is a document-store NoSQL database:

- **Documents** hold key/value fields (strings, timestamps, arrays, maps, etc.).
- **Collections** group documents; sub collections nest under documents.
- **Real-time & offline**: SDKs sync changes automatically and cache data when offline.

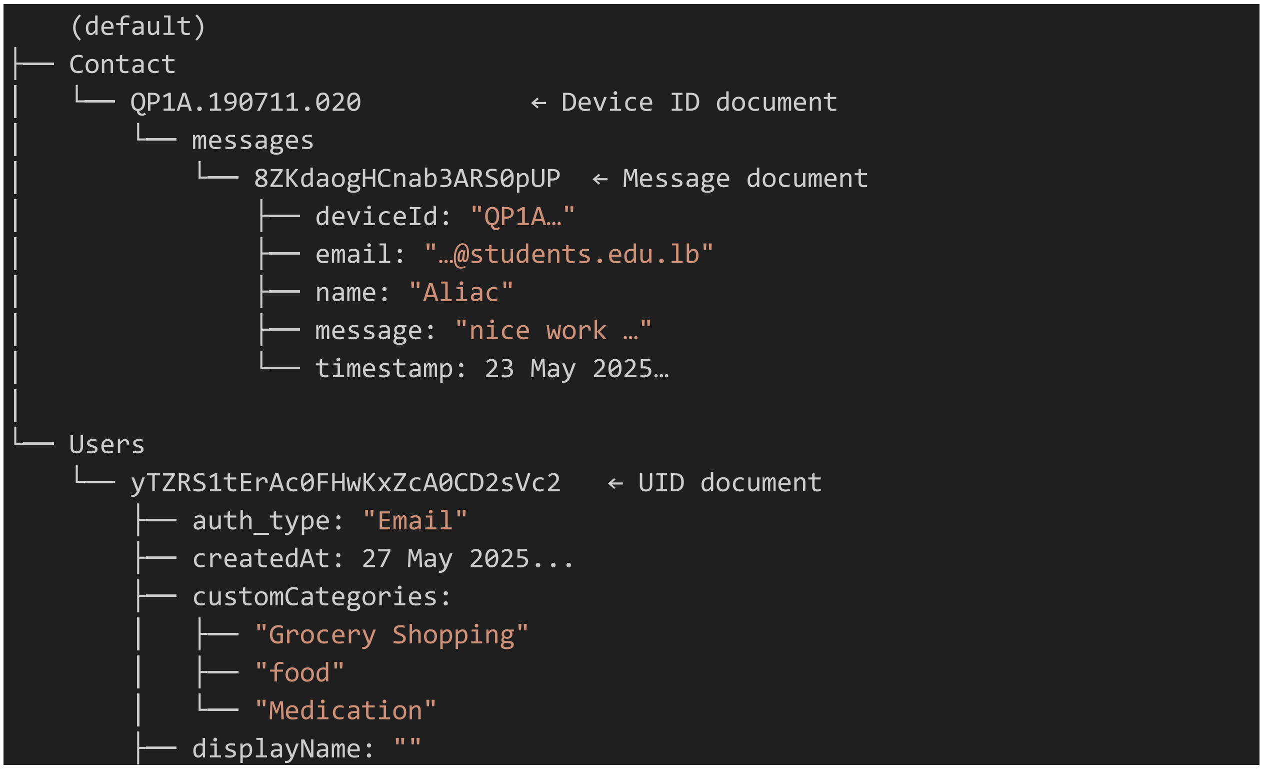
3. NoSQL Structure

Unlike SQL (tables & rows), Firestore is schema-less. You don't define tables or columns up front. Any collection or document path you write to is created on the fly:

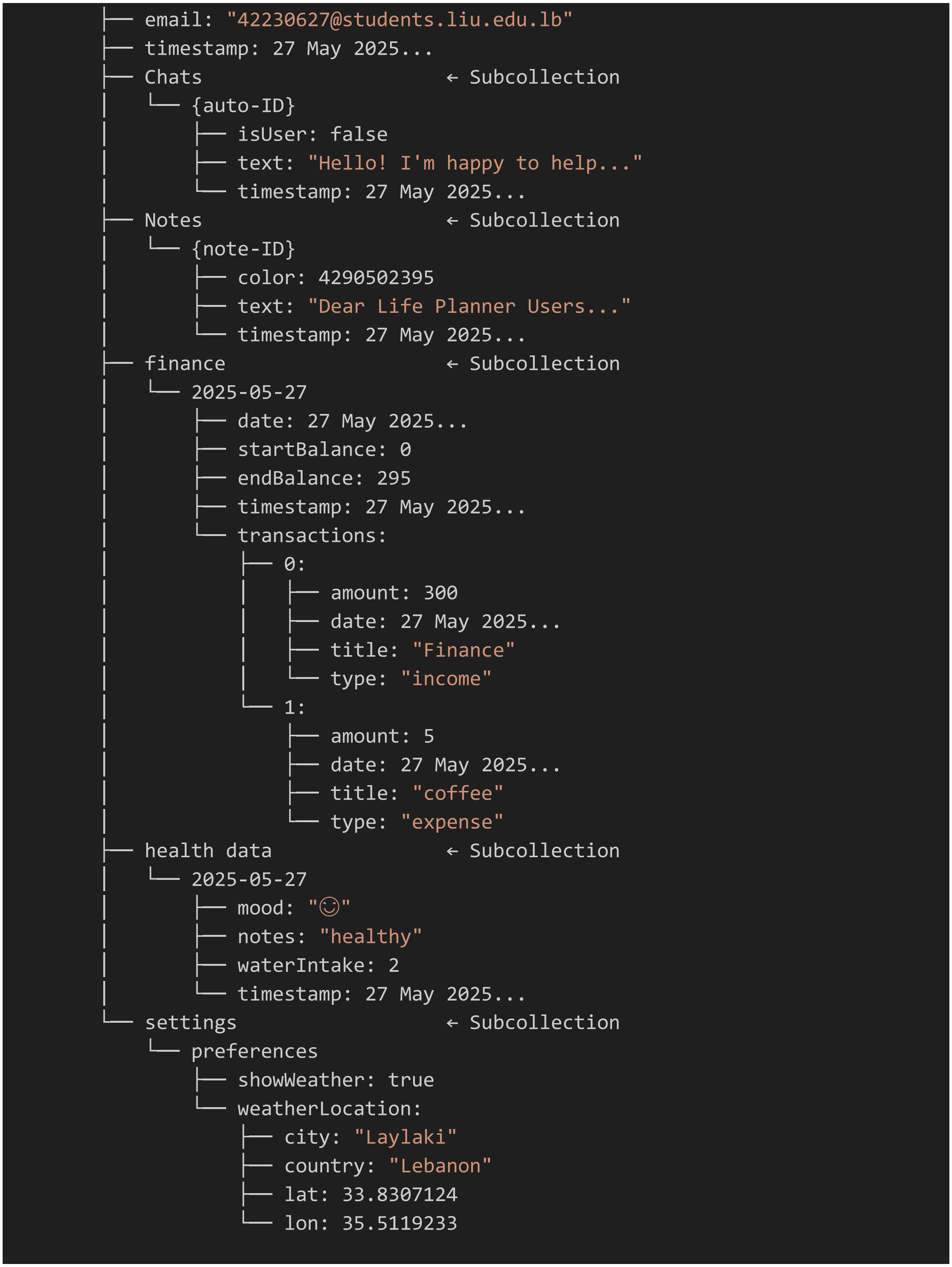
Example: writing a task under user “UID123”

```
FirebaseFirestore.instance
  .collection('users')        // → creates ‘users’ if needed
  .doc('UID123')              // → creates document UID123 if needed
  .collection('Tasks')        // → creates ‘Tasks’ subcollection
  .add({ title: 'Buy milk', dueDate: Timestamp.now() });
```

4. Database Tree Map



```
    (default)
├── Contact
│   └── QP1A.190711.020           ← Device ID document
│       └── messages
│           └── 8ZKdaogHCnab3ARS0pUP  ← Message document
│               ├── deviceId: "QP1A…"
│               ├── email: "…@students.edu.lb"
│               ├── name: "Aliac"
│               ├── message: "nice work …"
│               └── timestamp: 23 May 2025…
│
└── Users
    └── yTZRS1tErAc0FHwKxZcA0CD2sVc2   ← UID document
        ├── auth_type: "Email"
        ├── createdAt: 27 May 2025...
        ├── customCategories:
        │   ├── "Grocery Shopping"
        │   ├── "food"
        │   └── "Medication"
        ├── displayName: ""
```



```
├── email: "42230627@students.liu.edu.lb"
├── timestamp: 27 May 2025...
├── Chats              ← Subcollection
│ └── {auto-ID}
│   ├── isUser: false
│   ├── text: "Hello! I'm happy to help..."
│   └── timestamp: 27 May 2025...
├── Notes              ← Subcollection
│ └── {note-ID}
│   ├── color: 4290502395
│   ├── text: "Dear Life Planner Users..."
│   └── timestamp: 27 May 2025...
├── finance            ← Subcollection
│ └── 2025-05-27
│   ├── date: 27 May 2025...
│   ├── startBalance: 0
│   ├── endBalance: 295
│   ├── timestamp: 27 May 2025...
│   └── transactions:
│     ├── 0:
│     │ ├── amount: 300
│     │ ├── date: 27 May 2025...
│     │ ├── title: "Finance"
│     │ └── type: "income"
│     └── 1:
│       ├── amount: 5
│       ├── date: 27 May 2025...
│       ├── title: "coffee"
│       └── type: "expense"
├── health data        ← Subcollection
│ └── 2025-05-27
│   ├── mood: "😊"
│   ├── notes: "healthy"
│   ├── waterIntake: 2
│   └── timestamp: 27 May 2025...
└── settings           ← Subcollection
  └── preferences
    ├── showWeather: true
    └── weatherLocation:
      ├── city: "Laylaki"
      ├── country: "Lebanon"
      ├── lat: 33.8307124
      └── lon: 35.5119233
```

5. How Data Is Saved

- **Auto-creation**: Writing to any collection/document path auto-generates it—no migrations needed.
- **Subcollections**: Nest related data (messages, tasks, settings) under each user or device.
- **Timestamps**: Use Timestamp.now() for consistent server-side date/time.

6. Authentication & Security

Firebase Auth manages sign-in (Email, Google, etc.) and issues a unique **UID** per user.