KLANS — Finalized MongoDB Schema Document

1. User Model

```
const UserSchema = new mongoose.Schema({
  phone: { type: String, required: true, unique: true },
  name: { type: String },
  avatar: { type: String },
  referralCode: { type: String, unique: true },
  referredBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User', default:
null },
  onboardingStatus: { type: String, enum: ['invited', 'registered', 'active'],
default: 'invited' },
  klansJoined: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Klan' }],
  klansAdminOf: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Klan' }],
  role: { type: String, enum: ['user', 'admin', 'superadmin'], default:
'user' },
  inviteCodeUsed: { type: String },
  createdAt: { type: Date, default: Date.now },
  updatedAt: { type: Date }
});
```

Sample Add:

```
db.users.insertOne({
  phone: "+919876543210",
  name: "Sumit Kaila",
  avatar: "https://example.com/avatar.jpg",
  referralCode: "SUMIT123",
  onboardingStatus: "active",
  role: "user",
  createdAt: new Date()
});
```

2. OTP Model

```
const OTPSchema = new mongoose.Schema({
  phone: { type: String, required: true },
  otp: { type: String, required: true },
  otpExpiresAt: { type: Date, required: true },
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.otps.insertOne({
  phone: "+919876543210",
  otp: "123456",
  otpExpiresAt: new Date(Date.now() + 5*60*1000),
  createdAt: new Date()
});
```

---

3. Cluster Model

```
const ClusterSchema = new mongoose.Schema({
  name: { type: String, required: true },
  description: { type: String },
  icon: { type: String },
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  categories: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Category' }],
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.clusters.insertOne({
  name: "Mobility",
  description: "Communities related to transport and travel",
  icon: "https://example.com/mobility-icon.png",
  createdBy: ObjectId("64f123456789abcdef123456"),
  createdAt: new Date()
});
```

---

4. Category Model

```
const CategorySchema = new mongoose.Schema({
  name: { type: String, required: true },
  description: { type: String },
  icon: { type: String },
  clusterId: { type: mongoose.Schema.Types.ObjectId, ref: 'Cluster' },
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  uxSchema: { type: mongoose.Schema.Types.Mixed },
  isActive: { type: Boolean, default: true },
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.categories.insertOne({
  name: "Carpooling",
  description: "Pooling and ridesharing communities",
  icon: "https://example.com/carpooling-icon.png",
  clusterId: ObjectId("64f123456789abcdef123457"),
  createdBy: ObjectId("64f123456789abcdef123456"),
  uxSchema: {
    postType: "carpool",
    fields: [
      { name: "routeStart", type: "text", required: true },
      { name: "routeEnd", type: "text", required: true },
      { name: "date", type: "date", required: true },
      { name: "time", type: "time", required: true },
      { name: "price", type: "number", required: true }
    ],
    layout: "form",
    theme: { primaryColor: "#2C7BE5", secondaryColor: "#00D1B2" }
  },
  createdAt: new Date()
});
```

---

5. KLAN Model

```
const KlanSchema = new mongoose.Schema({
  name: { type: String, required: true },
  description: { type: String },
  location: {
    city: { type: String },
    state: { type: String },
    coordinates: { type: [Number], index: '2dsphere' }
  },
  categoryId: { type: mongoose.Schema.Types.ObjectId, ref: 'Category' },
  clusterId: { type: mongoose.Schema.Types.ObjectId, ref: 'Cluster' },
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  members: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
  admins: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
  privacy: { type: String, enum: ['public', 'invite-only'], default: 'public' },
  theme: { type: mongoose.Schema.Types.Mixed },
  uxSchemaOverride: { type: mongoose.Schema.Types.Mixed },
  inviteOnlyCode: { type: String },
  createdAt: { type: Date, default: Date.now },
  updatedAt: { type: Date }
});
```

Sample Add:

```
db.klans.insertOne({
  name: "Delhi-Chandigarh Carpool",
  description: "Daily commute carpool group",
  location: { city: "Delhi", state: "Delhi", coordinates: [77.1025, 28.7041] },
  categoryId: ObjectId("64f123456789abcdef123458"),
  clusterId: ObjectId("64f123456789abcdef123457"),
  createdBy: ObjectId("64f123456789abcdef123456"),
  members: [],
  admins: [ObjectId("64f123456789abcdef123456")],
  privacy: "invite-only",
  theme: { primaryColor: "#2C7BE5" },
  inviteOnlyCode: "DC123",
  createdAt: new Date()
});
```

---

6. Post Model

```
const PostSchema = new mongoose.Schema({
  clanId: { type: mongoose.Schema.Types.ObjectId, ref: 'Klan', required: true },
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },
  type: { type: String, enum: ['text', 'image', 'poll', 'event', 'carpool', 'custom'], default: 'text' },
  content: { type: String },
  media: [{ type: String }],
  meta: { type: mongoose.Schema.Types.Mixed },
  likes: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
  commentsCount: { type: Number, default: 0 },
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.posts.insertOne({
  clanId: ObjectId("64f123456789abcdef123459"),
  createdBy: ObjectId("64f123456789abcdef123456"),
  type: "carpool",
  content: "Carpool from Delhi to Chandigarh at 7 AM",
  media: [],
  meta: { routeStart: "Delhi", routeEnd: "Chandigarh", date: "2025-10-15",
time: "07:00", price: 300 },
  likes: [],
  commentsCount: 0,
  createdAt: new Date()
});
```

7. Comment Model

```
const CommentSchema = new mongoose.Schema({
  postId: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required:
true },
  text: { type: String, required: true },
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.comments.insertOne({
  postId: ObjectId("64f123456789abcdef123460"),
  createdBy: ObjectId("64f123456789abcdef123456"),
  text: "I'm interested, please share car details.",
  createdAt: new Date()
});
```

8. Event Model

```
const EventSchema = new mongoose.Schema({
  clanId: { type: mongoose.Schema.Types.ObjectId, ref: 'Klan', required: true },
  title: { type: String, required: true },
  description: { type: String },
  date: { type: Date, required: true },
  location: { type: String },
  maxParticipants: { type: Number },
  registeredUsers: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
  createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  createdAt: { type: Date, default: Date.now }
});
```

Sample Add:

```
db.events.insertOne({
  clanId: ObjectId("64f123456789abcdef123459"),
  title: "Delhi to Chandigarh Carpool Ride",
  description: "Weekly carpooling event",
  date: new Date("2025-10-15T07:00:00Z"),
  location: "Delhi",
  maxParticipants: 4,
  registeredUsers: [],
  createdBy: ObjectId("64f123456789abcdef123456"),
  createdAt: new Date()
});
```

This document contains the **final KLANS MongoDB schema** with OTP as a separate model, and sample add inserts for all models for reference.