MongoDB CRUD operation - Real-Time Quiz & Hackathon Platform

1. Users Collection

CREATE

```
db.Users.insertOne({
  name: "John Doe",
  email: "john@example.com",
  role: "Participant",
  profile: { bio: "Full Stack Dev", skills: ["JS", "Node.js"] },
  badges: ["Top Scorer"],
  createdAt: new Date()
})
```

READ

```
db.Users.find({ role: "Participant" }).pretty()
```

UPDATE

```
db.Users.updateOne({ email: "john@example.com" }, { $set: { role: "Moderator" } })
```

DELETE

```
db.Users.deleteOne({ email: "john@example.com" })
```

2. Quizzes Collection

CREATE

```
db.Quizzes.insertOne({
  title: "JavaScript Basics",
  description: "Test your JS knowledge",
  questions: [
```

```
    { questionText: "What is closure?", options: ["Scope", "Function"], answer:
"Function", type: "MCQ", difficulty: "Medium" }
  ],
  timeLimit: 20,
  startTime: new Date("2025-07-10T10:00:00Z"),
  endTime: new Date("2025-07-10T10:30:00Z"),
  createdBy: ObjectId("66b1f1a00000000000000002"),
  createdAt: new Date()
})
```

READ

```
db.Quizzes.find().pretty()
```

UPDATE

```
db.Quizzes.updateOne({ title: "JavaScript Basics" }, { $set: { timeLimit: 25 } })
```

DELETE

```
db.Quizzes.deleteOne({ title: "JavaScript Basics" })
```

3. Quiz Attempts Collection

CREATE

```
db.QuizAttempts.insertOne({
  quizId: ObjectId("66b1f1b00000000000000001"),
  userId: ObjectId("66b1f1a00000000000000001"),
  responses: [
    { questionId: ObjectId(), selectedOption: "Function", isCorrect: true, timeTaken: 8 }
  ],
  score: 1,
  completedAt: new Date()
})
```

READ

db.QuizAttempts.find().pretty()

UPDATE

db.QuizAttempts.updateOne({ quizId: ObjectId("66b1f1b00000000000000001") }, { $set: { score: 2 } })

DELETE

db.QuizAttempts.deleteOne({ userId: ObjectId("66b1f1a00000000000000001") })

4. Hackathons Collection

CREATE

```
db.Hackathons.insertOne({
 title: "AI for Good Hackathon",
 description: "Build AI for social impact",
 problems: [
   { title: "AI for Disaster Response", description: "Predict disasters", releaseTime: new Date("2025-08-01T09:00:00Z") }
 ],
 startTime: new Date("2025-08-01T09:00:00Z"),
 endTime: new Date("2025-08-03T18:00:00Z"),
 organizerId: ObjectId("66b1f1a00000000000000002"),
 teamsAllowed: true,
 teamSize: 4,
 createdAt: new Date()
})
```


READ

db.Hackathons.find().pretty()

UPDATE

```
db.Hackathons.updateOne({ title: "AI for Good Hackathon" }, { $set: { teamSize: 5 } })
```

DELETE

```
db.Hackathons.deleteOne({ title: "AI for Good Hackathon" })
```

5. Teams Collection

CREATE

```
db.Teams.insertOne({

  hackathonId: ObjectId("66b1f2d00000000000000001"),

  teamName: "Team Alpha",

  members: [ObjectId("66b1f1a00000000000000001"), ObjectId("66b1f1a00000000000000002")],

  createdAt: new Date()

})
```

READ

```
db.Teams.find().pretty()
```

UPDATE

```
db.Teams.updateOne({ teamName: "Team Alpha" }, { $push: { members: ObjectId("66b1f1a00000000000000003") } })
```

DELETE

```
db.Teams.deleteOne({ teamName: "Team Alpha" })
```

6. Submissions Collection

CREATE

```
db.Submissions.insertOne({

  hackathonId: ObjectId("66b1f2d00000000000000001"),

  teamId: ObjectId("66b1f4a00000000000000001"),

  submittedBy: ObjectId("66b1f1a00000000000000001"),
```

```
repositoryLink: "https://github.com/team-alpha/project",

documents: ["https://drive.google.com/file/d/abc123/view"],

presentation: "https://slides.com/team-alpha/project",

submittedAt: new Date(),

score: 85,

feedback: "Great job on real-time processing"

})
```

READ

```
db.Submissions.find().pretty()
```

UPDATE

```
db.Submissions.updateOne({ teamId: ObjectId("66b1f4a00000000000000001") }, { $set:
{ score: 90 } })
```

DELETE

```
db.Submissions.deleteOne({ teamId: ObjectId("66b1f4a00000000000000001") })
```

7. Leaderboard Collection

CREATE

```
db.Leaderboard.insertOne({

 eventType: "Hackathon",

 eventId: ObjectId("66b1f2d00000000000000001"),

 userId: ObjectId("66b1f1a00000000000000001"),

 score: 85,

 rank: 1,

 updatedAt: new Date()

})
```

READ

```
db.Leaderboard.find().pretty()
```

```
db.Leaderboard.updateOne({ userId: ObjectId("66b1f1a00000000000000001") }, { $set:
{ rank: 2 } })
```

DELETE

```
db.Leaderboard.deleteOne({ userId: ObjectId("66b1f1a00000000000000001") })
```

8. Notifications Collection

CREATE

```
db.Notifications.insertOne({

  userId: ObjectId("66b1f1a00000000000000001"),

  message: "Your hackathon starts in 1 hour",

  type: "Reminder",

  isRead: false,

  createdAt: new Date()

})
```

READ

```
db.Notifications.find({ userId: ObjectId("66b1f1a00000000000000001") }).pretty()
```

UPDATE

```
db.Notifications.updateOne({ userId: ObjectId("66b1f1a00000000000000001") }, { $set:
{ isRead: true } })
```

DELETE

```
db.Notifications.deleteOne({ message: "Your hackathon starts in 1 hour" })
```

9. Rewards Collection

CREATE

```
db.Rewards.insertOne({

  userId: ObjectId("66b1f1a00000000000000001"),
```

```
    eventId: ObjectId("66b1f1b00000000000000001"),

    eventType: "Quiz",

    rewardType: "Badge",

    rewardDetails: "Top Scorer in JavaScript Basics Quiz",

    issuedAt: new Date()

})
```

READ

```
db.Rewards.find().pretty()
```

UPDATE

```
db.Rewards.updateOne({ userId: ObjectId("66b1f1a00000000000000001") }, { $set: {
rewardType: "Certificate" } })
```

DELETE

```
db.Rewards.deleteOne({ rewardType: "Certificate" })
```