.Event Registration Portal – MongoDB Project Documentation.

Phase 1: Survey Design & Analysis

We began by designing a **Google Form** to capture participation details for our campus fest.

Steps Followed:

1. **Team Formation:** Formed a project team called **Data Divas** (4 members).

2. **Google Form Generation:** Designed a form with fields:
    - Timestamp
    - Full Name
    - College Email
    - Event Category
    - Event Game
    - Team Name (if any)
    - Team Members

3. **Form Circulation:** Shared the Google Form link with students via college groups.

4. **Response Collection:** Gathered responses over a fixed time window before the deadline.

5. **Saved Responses:** Downloaded responses as CSV from Google Form.

6. **Data Preparation:** Saved CSV in a dedicated folder and opened it in **VS Code** for inspection and cleaning (removed duplicates, fixed missing values).

7. **Database Preparation:**

    - Created a MongoDB database called festDB.

    - Created a collection called participants.

8. **Data Import:** Imported CSV into MongoDB using **mongoimport** command:

All form responses were inserted as documents in the participants collection

Phase 2: Case Study & Insights

- After importing data into MongoDB, we analyzed it using queries and aggregation pipelines.

Insights:

➔Group Dance was the most popular category (7 participants).

➔Carrom was the most popular game (4 participants).

➔Average team size was 3–4 members, confirming good team participation.

➔Found some missing team names → next iteration of Google Form will make Team Name a required field for group events.

Phase 3: MongoDB Data Model

- We used a **flat document model** because our CSV had a single response per row.
- Each row was converted into a single MongoDB document in the participants collection.

Sample Document Structure


```
{
  "_id": ObjectId("6504b1b2f8d3a67c89012345"),
  "Timestamp": "2025-09-15T18:23:06Z",
  "Full name": "Shweta",
  "College Email": "2024.shwetas@isu.ac.in",
  "event-Category": "Group Dance",
  "event-Games": "Chess",
  "Team name(if any)": "tralala",
  "Team members": 3
}
```

Schema Explanation:

Field Name	Type	Description
1.Timestamp	Date	Auto-Generated Submission timestamp
2.Full name	String	Participant’s name
3.College email	String	Unique email
4.Event-category	String	Category of participation
5.Event-Games	String	Games chosen(if any)
6.Team name(if any)	String	Team’s name for group events
7.Team Members	Number	Number of participants in the team

Phase 4: Querying & Results

- Once imported to MongoDB, sample data looks like this (using insertMany()):

```
db.registrations.insertMany([
  {
    fullName: "Payal Kunwar",
    collegeEmail: "abc@gmail.com",
    registrations: [
      { eventCategory: "Solo Singing", eventGame: null,
teamName: null, teamMembers: 1, timestamp:
ISODate("2025-09-15T18:21:58Z") }
    ]
```

```
  },
  {
    fullName: "Shweta",
    collegeEmail: "2024.shwetas@isu.ac.in",
    registrations: [
      { eventCategory: "Group Dance", eventGame: "Chess",
teamName: "tralala", teamMembers: 3, timestamp:
ISODate("2025-09-15T18:23:06Z") }
    ]}])
```

- After importing, we ran queries to gain insights:

Example Queries:

1. View all participants in Group Dance

```
db.participants.find({ "event-Category": "Group Dance" })
```

2. Count participants per event category

```
db.participants.aggregate([
  { $group: { _id: "$event-Category", count: { $sum: 1 } } }
])
```

3. Find teams with more than 3 members

```
db.participants.find({ "Team members": { $gt: 3 } })
```

4. List distinct event games

```
db.participants.distinct("event-Games")
```

Phase 5: Results & Learnings

1. Successfully imported 16 participant records into MongoDB.

2. Learned how to use mongoimport to migrate CSV → MongoDB.

3. Practiced writing basic queries (find, aggregate, distinct) to analyze data.

- **Identified areas of improvement:**
  1.Add validation for Team members count (must be >1 for group events).
  2.Implement participation caps using MongoDB aggregation (count & compare with max limit).

Conclusion:

This project helped us implement end-to-end data collection, cleaning, and storage in MongoDB, while also performing analytics using queries and pipelines.

GitHub Link:

https://vrutti88.github.io/MongoDB-CollegeFest-Project/html/index.html