**Data Management Plan (Iteration 2)**

**Link to WIKI page:** <https://github.com/Ramakotireddy9505/GDP-Team04-foodiefinds/wiki/Data-Management-Plan-(Iteration-2)>

**1.Users Collection:**

• Each document in the "users" collection represents a user.

• **Fields:** UserID, Name, Email, Password, Role, Preferences.

**Sample document of above collection :**

{

"UserID": "",

"Name": "",

"Email": "",

"Password": "",

"Role": "",

"Preferences": ""

}

**2.Restaurants Collection:**

• Each document in the "restaurants" collection represents a restaurant.

• **Fields:** RestaurantID, Name, Location, Cuisine, PriceRange, AverageRating, OpenHours, MenuItems, TableAvailability, Photos.

**Sample document in collection:**

{

"RestaurantID": "",

"Name": "",

"Location": "",

"Cuisine": "",

"PriceRange": "",

"AverageRating": "",

"OpenHours": "",

"MenuItems": "",

"TableAvailability": "",

"Photos": ""

}

**3.Reservations Collection:**

• Each document in the "reservations" collection represents a reservation.

• **Fields:** ReservationID, UserID, RestaurantID, Date, Time, PartySize.

**Sample document of above collection:**

{

"ReservationID": "",

"UserID": "",

"RestaurantID": "",

"Date": "",

"Time": "",

"PartySize": ""

}

**4.Reviews Collection:**

• Each document in the "reviews" collection represents a review.

• **Fields:** ReviewID, UserID, RestaurantID, Rating, ReviewText, Date.

**Sample document of above collection:**

{

"ReviewID": "",

"UserID": "",

"RestaurantID": "",

"Rating": "",

"ReviewText": "",

"Date": ""

}

**Initial Plans to Secure Data:**

**Access Restriction:** Use role-based access control (RBAC) to restrict access depending on user role. Admins will oversee restaurant and review features, while regular users will concentrate on restaurant discovery and reservations.

**Mapping of Functional Requirements to Data Storage:**

**System Registration and Authentication:** Store user registration and authentication data in the Users collection, which includes fields like name, email, and password. RBAC ensures that personalized features are only accessible to verified users with the proper roles.

**Restaurant Exploration and Management:** Maintain restaurant data in the Restaurants collection, such as name, location, cuisine, ratings, and menu items. Admins can edit this information through the admin interface, which is synced with the Restaurants and Reviews collections.

**Integration with Restaurant Data Sources:** Fetch real-time changes on restaurant availability and menus from external APIs and save them to the Restaurants collection.

**Personalized Restaurant Search:** Use user preferences from the Users collection to make personalized recommendations. Search engines match user criteria to restaurant data in the Restaurants collection to produce relevant results.

**Admin Review Management:** Securely store admin-contributed reviews in the Reviews collection, which are tied to specific restaurant and user IDs.

**Sorting of Restaurant Reviews:** Restaurant reviews can be saved in the Reviews collection and sorted according to user preferences.

**Restaurant Reservation Management:** The Reservations collection contains reservation information such as the restaurant, date, time, and number of guests. To streamline the booking process, select available restaurants and reservation options from the Reservations collection.