### Slide 1: Team Information

- \*\*Title: Team Information\*\*

- \*\*Content:\*\*

- \*\*Team Member 1:\*\*

- Name: [Name]

- SBU\_ID: [ID]

- Role: [Role]

- \*\*Team Member 2:\*\*

- Name: [Name]

- SBU\_ID: [ID]

- Role: [Role]

- \*\*Team Member 3:\*\*

- Name: [Name]

- SBU\_ID: [ID]

- Role: [Role]

### Slide 2: Team Information (continued)

- \*\*Content:\*\*

- \*\*Team Member 4:\*\*

- Name: [Name]

- SBU\_ID: [ID]

- Role: [Role]

- \*\*Team Member 5:\*\*

- Name: [Name]

- SBU\_ID: [ID]

- Role: [Role]

### Slide 3: Design (ER/EER Model)

- \*\*Title: ER/EER Model\*\*

- \*\*Content:\*\*

- Insert the ER/EER model diagram image here (from the provided PNG file).

### Slide 4: Entities and Attributes - Users

- \*\*Title: Entities and Attributes - Users\*\*

- \*\*Content:\*\*

- \*\*Entity: Users\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- name: VARCHAR(255)

- phone\_number: VARCHAR(15)

- email: VARCHAR(255), Unique

- otp: VARCHAR(10)

- is\_verified: BOOLEAN

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- email must be unique

- \*\*Relationships:\*\*

- Users have relationships with Orders, Feedbacks, Sessions, and Carts.

### Slide 5: Entities and Attributes - Restaurants

- \*\*Title: Entities and Attributes - Restaurants\*\*

- \*\*Content:\*\*

- \*\*Entity: Restaurants\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- name: VARCHAR(255), Not Null

- logo\_url: VARCHAR(255)

- opening\_time: TIMESTAMP

- closing\_time: TIMESTAMP

- status: TEXT, Check ('open', 'closed')

- operating\_days: JSONB

- phone\_number: VARCHAR(15)

- email: VARCHAR(255), Unique

- encrypted\_password: TEXT, Not Null

- password\_hash: TEXT, Not Null

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- email must be unique, status must be 'open' or 'closed'

- \*\*Relationships:\*\*

- Restaurants have relationships with Menu, Courses, Food Items, Orders, Feedbacks, Sessions, Carts, and Tables.

### Slide 6: Entities and Attributes - Menu

- \*\*Title: Entities and Attributes - Menu\*\*

- \*\*Content:\*\*

- \*\*Entity: Menu\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- name: VARCHAR(255)

- restaurant\_id: UUID

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- restaurant\_id references restaurants(id)

- \*\*Relationships:\*\*

- Menu is related to Food Items.

### Slide 7: Entities and Attributes - Courses

- \*\*Title: Entities and Attributes - Courses\*\*

- \*\*Content:\*\*

- \*\*Entity: Courses\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- name: VARCHAR(255)

- restaurant\_id: UUID

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- restaurant\_id references restaurants(id)

- \*\*Relationships:\*\*

- Courses are related to Food Item Courses.

### Slide 8: Entities and Attributes - Food Items

- \*\*Title: Entities and Attributes - Food Items\*\*

- \*\*Content:\*\*

- \*\*Entity: Food Items\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- custom\_id: VARCHAR(50)

- name: VARCHAR(255)

- menu\_id: UUID

- restaurant\_id: UUID

- description: TEXT

- price: VARCHAR(50)

- food\_img\_url: VARCHAR(255)

- dietary\_restriction: TEXT, Check ('veg', 'non\_veg', 'egg')

- is\_featured: BOOLEAN

- spicy\_rating: INT

- status: TEXT, Check ('in\_stock', 'out\_of\_stock')

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- dietary\_restriction must be 'veg', 'non\_veg', or 'egg'; status must be 'in\_stock' or 'out\_of\_stock'

- \*\*Relationships:\*\*

- Food Items are related to Menu, Restaurant, Order Items, and Food Item Courses.

### Slide 9: Entities and Attributes - Orders

- \*\*Title: Entities and Attributes - Orders\*\*

- \*\*Content:\*\*

- \*\*Entity: Orders\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- custom\_order\_id: VARCHAR(12)

- restaurant\_id: UUID

- user\_id: UUID

- total\_amount: VARCHAR(50)

- table\_id: UUID

- invoice\_url: VARCHAR(255)

- is\_paid: BOOLEAN

- status: TEXT, Check ('order\_placed', 'confirmed', 'in\_process', 'completed', 'deleted')

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- status must be 'order\_placed', 'confirmed', 'in\_process', 'completed', or 'deleted'

- \*\*Relationships:\*\*

- Orders are related to Restaurant, User, Table, Order Items, and Feedbacks.

### Slide 10: Entities and Attributes - Order Items

- \*\*Title: Entities and Attributes - Order Items\*\*

- \*\*Content:\*\*

- \*\*Entity: Order Items\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- order\_id: UUID

- food\_item\_id: UUID

- amount: VARCHAR(50)

- quantity: INT

- status: TEXT, Check ('order\_placed', 'confirmed', 'in\_process', 'completed', 'deleted')

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- status must be 'order\_placed', 'confirmed', 'in\_process', 'completed', or 'deleted'

- \*\*Relationships:\*\*

- Order Items are related to Orders and Food Items.

### Slide 11: Entities and Attributes - Tables

- \*\*Title: Entities and Attributes - Tables\*\*

- \*\*Content:\*\*

- \*\*Entity: Tables\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- name: VARCHAR(255)

- table\_no: VARCHAR(50)

- restaurant\_id: UUID

- status: TEXT, Check ('occupied', 'available', 'reserved')

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Constraints:\*\*

- status must be 'occupied', 'available', or 'reserved'

- \*\*Relationships:\*\*

- Tables are related to Restaurant, Orders, and Sessions.

### Slide 12: Entities and Attributes - Feedbacks

- \*\*Title: Entities and Attributes - Feedbacks\*\*

- \*\*Content:\*\*

- \*\*Entity: Feedbacks\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- text: TEXT

- restaurant\_id: UUID

- star\_rating: FLOAT

- user\_id: UUID

- order\_id: UUID

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- deleted\_at: TIMESTAMP

- \*\*Relationships:\*\*

- Feedbacks are related to Restaurant, User, and Orders.

### Slide 13: Entities and Attributes - Sessions

- \*\*Title: Entities and Attributes - Sessions\*\*

- \*\*Content:\*\*

- \*\*Entity: Sessions\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- session\_id: UUID

- restaurant\_id: UUID

- table\_id: UUID

- additional\_note: TEXT

- user\_id: UUID

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- \*\*Relationships:\*\*

- Sessions are related to Restaurant, Table, and User.

### Slide 14: Entities and Attributes - Carts

- \*\*Title: Entities and Attributes - Carts\*\*

- \*\*Content:\*\*

- \*\*Entity: Carts\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- session\_id: UUID

- restaurant\_id: UUID

- table\_id: UUID

-

additional\_note: TEXT

- user\_id: UUID

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- \*\*Relationships:\*\*

- Carts are related to Restaurant, Table, User, and Cart Items.

### Slide 15: Entities and Attributes - Cart Items

- \*\*Title: Entities and Attributes - Cart Items\*\*

- \*\*Content:\*\*

- \*\*Entity: Cart Items\*\*

- \*\*Attributes:\*\*

- id: UUID, Primary Key

- cart\_id: UUID

- food\_item\_id: UUID

- instruction\_note: TEXT

- quantity: INT

- created\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- updated\_at: TIMESTAMP, Default CURRENT\_TIMESTAMP

- \*\*Relationships:\*\*

- Cart Items are related to Carts and Food Items.

### Slide 16: Interesting Question 1

- \*\*Title: Interesting Question 1\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* List all the restaurants that are currently open, along with their opening and closing times.

- \*\*SQL Query:\*\*

```sql

SELECT restaurants.name AS restaurant\_name, restaurants.opening\_time, restaurants.closing\_time

FROM restaurants

WHERE restaurants.status = 'open';

```

### Slide 17: Interesting Question 2

- \*\*Title: Interesting Question 2\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Which food items have been ordered the most, and how many times have they been ordered?

- \*\*SQL Query:\*\*

```sql

SELECT food\_items.name AS food\_item\_name, COUNT(order\_items.id) AS order\_count

FROM order\_items

JOIN food\_items ON order\_items.food\_item\_id = food\_items.id

GROUP BY food\_items.name

ORDER BY order\_count DESC;

```

### Slide 18: Interesting Question 3

- \*\*Title: Interesting Question 3\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* List the names of users who have given feedback with a star rating of 5 for any restaurant.

- \*\*SQL Query:\*\*

```sql

SELECT DISTINCT users.name

FROM feedbacks

JOIN users ON feedbacks.user\_id = users.id

WHERE feedbacks.star\_rating = 5;

```

### Slide 19: Interesting Question 4

- \*\*Title: Interesting Question 4\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Find the average star rating for each restaurant.

- \*\*SQL Query:\*\*

```sql

SELECT restaurants.name AS restaurant\_name, AVG(feedbacks.star\_rating) AS average\_rating

FROM feedbacks

JOIN restaurants ON feedbacks.restaurant\_id = restaurants.id

GROUP BY restaurants.name;

```

### Slide 20: Interesting Question 5

- \*\*Title: Interesting Question 5\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Retrieve all the orders along with their associated user name, restaurant name, and total amount, filtered by a specific date range.

- \*\*SQL Query:\*\*

```sql

SELECT orders.id AS order\_id, users.name AS user\_name, restaurants.name AS restaurant\_name, orders.total\_amount

FROM orders

JOIN users ON orders.user\_id = users.id

JOIN restaurants ON orders.restaurant\_id = restaurants.id

WHERE orders.created\_at BETWEEN '2024-01-01' AND '2024-12-31';

```

### Slide 21: Interesting Question 6

- \*\*Title: Interesting Question 6\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Get the total sales amount for each restaurant by summing the total\_amount from orders.

- \*\*SQL Query:\*\*

```sql

SELECT restaurants.name AS restaurant\_name, SUM(orders.total\_amount::numeric) AS total\_sales

FROM orders

JOIN restaurants ON orders.restaurant\_id = restaurants.id

GROUP BY restaurants.name;

```

### Slide 22: Interesting Question 7

- \*\*Title: Interesting Question 7\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Which tables in which restaurants are currently occupied?

- \*\*SQL Query:\*\*

```sql

SELECT restaurants.name AS restaurant\_name, tables.name AS table\_name, tables.table\_no

FROM tables

JOIN restaurants ON tables.restaurant\_id = restaurants.id

WHERE tables.status = 'occupied';

```

### Slide 23: Interesting Question 8

- \*\*Title: Interesting Question 8\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Find the total amount spent by each user across all their orders.

- \*\*SQL Query:\*\*

```sql

SELECT users.name AS user\_name, SUM(CAST(orders.total\_amount AS NUMERIC)) AS total\_spent

FROM orders

JOIN users ON orders.user\_id = users.id

GROUP BY users.name

ORDER BY total\_spent DESC;

```

### Slide 24: Interesting Question 9

- \*\*Title: Interesting Question 9\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* List the names of the top 5 most expensive food items available.

- \*\*SQL Query:\*\*

```sql

SELECT food\_items.name AS food\_item\_name, CAST(food\_items.price AS NUMERIC) AS item\_price

FROM food\_items

ORDER BY item\_price DESC

LIMIT 5;

```

### Slide 25: Interesting Question 10

- \*\*Title: Interesting Question 10\*\*

- \*\*Content:\*\*

- \*\*Question:\*\* Show the restaurant and user details for orders that have not been paid.

- \*\*SQL Query:\*\*

```sql

SELECT orders.id AS order\_id, restaurants.name AS restaurant\_name, users.name AS user\_name, orders.total\_amount

FROM orders

JOIN restaurants ON orders.restaurant\_id = restaurants.id

JOIN users ON orders.user\_id = users.id

WHERE orders.is\_paid = FALSE;