**KFF’ 2024 MANAGEMENT API**

**Kirtan department**

**1. User Database (User Table)**

The User table contains the details of registered users. It handles user information, balance tracking, NFC card information, and more.

**Fields:**

* **Slot ID**: Auto-incremented field to uniquely identify each slot.
* **User ID**: Unique ID assigned to each user.
* **Full Name**: The full name of the user.
* **Email**: Email address of the user.
* **Phone Number**: The phone number associated with the user.
* **NFC ID**: The unique NFC card ID tied to the user.
* **Balance**: Current balance of the user.
* **Last Transaction**: The time of the last transaction carried out by the user.
* **Time In**: The user's entry time.
* **Time Out**: The user's exit time.

**2. Vendor Database (Vendor Table)**

The Vendor table tracks vendor details, including balance and last transaction information.

**Fields:**

* **Vendor ID**: Unique ID assigned to each vendor.
* **Vendor Name**: Name of the vendor.
* **Vendor Phone Number**: Vendor's contact number.
* **Vendor Balance**: The vendor’s current balance.
* **Vendor Last Transaction**: The time and date of the last transaction made by the vendor.

**3. Transaction Database (Transaction Table)**

The Transaction table tracks all the transactions carried out in the system, including POS transactions and top-up operations.

**Fields:**

* **Transaction ID**: A unique 8-digit alphanumeric ID for each transaction.
* **NFC ID**: The NFC ID of the user involved in the transaction.
* **Name**: Name of the customer or user involved.
* **Vendor**: The vendor involved in the transaction.
* **Transaction Amount**: The amount of money involved in the transaction.
* **Status**: The status of the transaction (e.g., 'processed', 'failed').
* **Date and Time**: The date and time when the transaction occurred.
* **Type of Transaction**: Describes the nature of the transaction (e.g., 'POS', 'Top-up').

**API Routes Documentation**

**1. POST /assignnfc**

This route assigns the NFC card details for a specific user and sets the balance to 5 credits by default.

**Request:**

* **Method**: POST
* **Data**:

{

"uid": "user\_nfc\_id",

"nfc\_card": "new\_nfc\_id"

}

**Process:**

* Locates user based on ID number provided in header and updated said user with a nfc card id, which is in the body.

**Example Request:**

-

**Response:**

{

"message": "NFC card updated successfully"

}

**2. POST /process\_transaction**

This route handles the transaction process by checking if a user has enough balance and deducting the requested amount.

**Request:**

* **Method**: POST
* **Data**:

{

"vendor\_id": "vendor\_001",

"nfc\_id": "12345678",

"amount": 10

}

**Process:**

* The server generates an 8-digit alphanumeric Transaction ID.
* Looks up the user using the NFC ID and checks their current balance.
* Deducts the requested amount if the user has enough balance.
* Updates the User and Transaction tables.
* Updates the Vendor table if the transaction is successful.

**Example Request:**

curl -X POST http://127.0.0.1:5000/process\_transaction -d '{"vendor\_id": "vendor\_001", "nfc\_id": "12345678", "amount": 10}' -H "Content-Type: application/json"

**Example Response (successful):**

{

"transaction\_id": "A1B2C3D4",

"user": {

"name": "John Doe",

"new\_balance": 90

},

"status": "Transaction Successful"

}

**Example Response (failed due to insufficient balance):**

json

Copy code

{

"message": "Insufficient balance",

"current\_balance": 5

}

**3. POST /add\_vendor**

This route adds a new vendor to the Vendor table.

**Request:**

* **Method**: POST
* **Data**:

{

"vendor\_name": "Vendor 001",

"vendor\_phone": "987654321"

}

**Process:**

* Adds a new vendor entry to the Vendor table.

**Example Request:**

curl -X POST http://127.0.0.1:5000/add\_vendor -d '{"vendor\_name": "Vendor 001", "vendor\_phone": "987654321"}' -H "Content-Type: application/json"

**Example Response:**

{

"message": "Vendor added successfully",

"vendor\_id": 1

}

**4. GET /get\_balance/<user\_id>**

This route retrieves the current balance of a user.

**Request:**

* **Method**: GET
* **Parameters**: user\_id

**Example Request:**

curl -X GET http://127.0.0.1:5000/get\_balance/1

**Example Response:**

{

"user\_id": 1,

"balance": 100

}

**5. GET /vendor\_balance/<vendor\_id>**

This route checks the balance of a specific vendor.

**Request:**

* **Method**: GET
* **Parameters**: vendor\_id

**Example Request:**

curl -X GET http://127.0.0.1:5000/vendor\_balance/1

**Example Response:**

{

"vendor\_id": 1,

"balance": 500

}

**6. POST /topup**

This route processes a top-up to a user’s account.

**Request:**

* **Method**: POST
* **Data**:

{

"topup\_source": "source\_name",

"nfc\_id": "12345678",

"amount": 20

}

**Process:**

* The server finds the user based on the NFC ID.
* Adds the specified top-up amount to the user's balance.
* Logs the transaction in the Transaction table as a top-up.

**Example Request:**

curl -X POST http://127.0.0.1:5000/topup -d '{"topup\_source": "ATM", "nfc\_id": "12345678", "amount": 20}' -H "Content-Type: application/json"

**Example Response:**

{

"transaction\_id": "A7B6C5D4",

"user": {

"name": "John Doe",

"new\_balance": 120

},

"status": "Top-up Successful"

}

**7. POST /create\_user**

This route allows you to create a new user and add them to the User table in the database.

**Request:**

* **Method**: POST
* **Data**:

{

"name": "John Doe",

"email": "johndoe@example.com",

"phone\_number": "1234567890"

}

**Process:**

* The server receives the user data (name, email, phone number).
* A new user entry is added to the User table.
* The system assigns a unique User ID and default values for other fields (e.g., Balance is initialized at 5 credits).

**Example Request:**

curl -X POST http://127.0.0.1:5000/create\_user -d '{"name": "John Doe", "email": "johndoe@example.com", "phone\_number": "1234567890"}' -H "Content-Type: application/json"

**Example Response:**

{

"message": "User created successfully",

"user\_id": 1

}

**8. GET /fetch\_user**

This route retrieves all users associated with a specific phone number from the User table.

**Request:**

* **Method**: GET
* **Parameters**: phone\_number

**Example Request:**

curl -X GET http://127.0.0.1:5000/fetch\_user?phone\_number=1234567890

**Process:**

* The server retrieves all users who share the given phone number.
* It returns a list of user records, each including details like User ID, Name, Email, Phone Number, and NFC ID.

**Example Response:**

{

"users": [

{

"user\_id": 1,

"name": "John Doe",

"email": "johndoe@example.com",

"phone\_number": "1234567890",

"nfc\_card": "NFC\_1234"

},

{

"user\_id": 2,

"name": "Jane Doe",

"email": "janedoe@example.com",

"phone\_number": "1234567890",

"nfc\_card": "NFC\_5678"

}

]

}