# Bank Account

## Instruction:

- Write the class **BankAccount** according to the following specifications. To simulate basic banking operations such as depositing and withdrawing funds through an account. The simulation includes recording each transaction with details like transaction type, amount, and timestamp. It aims to provide methods for account interaction and transaction history review.

**Class BankAccount ::**

**Constructor**
- **constructor(accountNumber)**
  - **Parameters**: **name** type: string, **position** type: string
  - Initializes a new **BankAccount** instance with a given account number. It sets the initial balance to zero and prepares an empty array for storing transaction details.

**Properties**
- **accountNumber**: A unique identifier for the account, type: String.
- **balance**: The current balance of the account, type: number.
- **transactions**: An array to store details of each transaction. Each transaction object includes **type**, **amount**, and **timestamp**, type: array of object.

  {

  **Id**: id is running number create by transactions.length + 1, type: number

  **type**: type, type: String ex: "deposit", "withdrawal",

  **amount**: **amount**, type: number,

  **timestamp**: date, type: date

  }

**Methods**
- **deposit(amount)**
  - **Parameters**: **amount** type: number
  - Increases the account balance by the specified amount if it's positive. **Records the deposit transaction** with its details. **Returns** the **new balance.**
- **withdraw(amount)**
  - **Parameters**: **amount** type: number
  - Decreases the account balance by the specified amount if there are sufficient funds and the amount is positive. **Records the withdrawal**

**transaction** with its details. **Returns** the **new balance** or **null** if insufficient funds.

- **getBalance()**
  - **Parameters**: -
  - **Returns** the current balance of the account.
- **getAccountNumber()**
  - **Parameters**: -
  - **Returns** the unique account number associated with the account.
- **getTransactions()**
  - **Parameters**: -
  - **Returns** the array of recorded transactions, including details such as type, amount, and timestamp for each transaction.
- **getTransaction(id)**
  - **Parameters**: **id** type: number
  - Retrieves a specific transaction by its ID from the transactions array. **Returns** the transaction details or **null** if not found.
- **recordTransaction(type, amount)**
  - **Parameters**: **type** type: string**, amount** type: number
  - Records details of a transaction, including its **type** ('deposit' or 'withdrawal'), **amount**, and the current **timestamp**. This method is usually called internally by the **deposit** and **withdraw** methods.

*For Module 1.1

- **transferFunds(targetAccountNumber, amount)**
  - **Parameters**: **targetAccountNumber** type: string, **amount** type: number
  - Transfers the specified amount from this account to another account identified by the target account number, if funds allow. Deducts the amount from this account and adds it to the target account. Returns true if the transfer was successful, false if not enough funds or targetAccountNumber not exists.

```
class BankAccount {
}

// Example Usage
const myAccount = new BankAccount("123456789");
console.log(myAccount.deposit(100)); // Deposit $100
console.log(myAccount.withdraw(50)); // Withdraw $50
console.log(myAccount.getBalance()); // Get the current balance
console.log(myAccount.getTransactions()); // Get all transactions
```