**Lab Assignment: Build ASP.NET Core Web API for Banking Application CRUD**

**Objective:**

Create a RESTful ASP.NET Core Web API that allows performing CRUD operations on customer bank accounts.

**Use Case Description:**

You are developing a backend API for a banking system. The system should manage customer bank accounts with the following operations:

1. Create a new bank account
2. Retrieve all bank accounts or by account number
3. Update existing account details
4. Delete an account (soft delete if possible)

**Entity: BankAccount**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| AccountId | int (PK) | Unique identifier for the account |
| CustomerName | string | Name of the account holder |
| AccountNumber | string | Unique account number |
| AccountType | string | e.g., Savings, Current |
| Balance | decimal | Current balance |
| CreatedDate | DateTime | Account creation date |
| IsActive | bool | Indicates whether the account is active |

**🔧 Tasks to Implement:**

**1. Setup Project**

* Create an **ASP.NET Core Web API** project.
* Use **ADO.NET**.
* Configure **SQL Server** **Database**.

**2. Create the Model and DbContext**

* Create the BankAccount entity class.
* Add BankingDbContext with DbSet<BankAccount>.

**3. Implement CRUD Endpoints in Controller**

* POST /api/accounts – Create a new bank account
* GET /api/accounts – Get all accounts
* GET /api/accounts/{id} – Get account by ID
* PUT /api/accounts/{id} – Update account details
* DELETE /api/accounts/{id} – Delete (or deactivate) an account

**4. Add Basic Validation**

* Ensure AccountNumber is unique.
* Ensure CustomerName and AccountType are required.
* Ensure Balance is non-negative.

**5. Test API with Swagger or Postman**

* Use Swagger UI to test all CRUD operations.
* Verify proper responses (200 OK, 201 Created, 404 Not Found, 400 Bad Request, etc.).