**Fibonacci Numbers**

**Frontend: frontend-fibo**

Technology: Angular 15

Component:

1. Input: Consists of an Input text box for number and Submit button to get the data from API. OnSubmit() method is used to check whether the integer value is present or not. If present, then it will call the service by passing the N value. If not present, it will throw an alert.
2. Output: This page will show the heading and the result of API calls along with the Back button which will redirect you to the input or main page.
3. Fibo Service: We have used the service to make the API call to “http://localhost:8080/ fiboNo/{input}”. Get the response and error. If there is a response it uses the router to navigate to the output page and save the result and input value to the browser’s history. If an error occurred it shows the alert message and throws a new error.

**Backtend: backendfibo**

Technology: Java, Spring Boot, JPA, REST, H2 Database,

Component:

1. **FiboController.java**: It is a Rest Controller where REST APIs are created for the application. fiboNo and getAllFiboData endpoints are created to get the first N Fibonacci numbers and get all data present in the Database.
2. **FiboDataModel.java**: It is a Data Model create to connect with H2 Database with column id (Auto Generated), Index (to Store N value), value (Calculated value) with the getter-setter methods, and constructor to create an object.
3. **FiboDataRepository.java**: Created an interface that implements the JPA repository to handle the database interaction with the Fibo data model. Add the count() method to get the number of records and findByFiboIndexLessThanEqualOrderByFiboIndex to get the record less than equal to the N value.
4. **FiboDataService.java**: It is a service file where all the business logic or operation is going to perform. It uses a Repository object to operate.
   1. **getFiboNo()** method will check if the data is present or not. If not present it will add the 0 and 1 in the database as the initial value for calculation. If there is data in the DB, it will fetch the data less than by index as the index store the N value. We have extracted the value from records and stored it in the new list. If the size of the list is equal to N, we will return the list. If not, We will pass the value to findFiboNo() method to compute the value.
   2. **findFiboNo()** method takes the N and Series of values. It will calculate the Fibonacci numbers by taking the end two values in the list. This value will be assigned to respective variables and for loop will be executed from the size of the series +1 till the N value and calculates the value to store in the result along with the old series data. Will calculating we will create the Fibo Data Model objects list for new calculated values and save at the end using the saveAllFiboData() method and will return the result with all values.
   3. **saveAllFiboData()** method is the transactional method because it going to store all the data in the database if not present.
   4. **getAllFiboData()**  method will give all the data in the Database.