My name is Subhashish Behera. I have chosen task 2 to create a Web Application which allows the operator to manage the customer and booking , and also created a database in the backend to store the data.

[Working Video of the task](https://drive.google.com/file/d/1x5bHbPRgmPoAmYP53I6zDusHpblyh7U-/view?usp=sharing)

Tech Stack Used - Appian which has its own cloud database (MariaDB) and frontend (interface, site and portal)

Approach - First I created 4 database ( via Record Type )

1.Location

2. Booking

3. Customer

4. Drone Shot

I chose record type as it makes relationships between tables easy and it contains record actions which were used to perform CRUD operations.

Then I made 4 data types as it is easy to query entities (logics) and added all of them to a data store to connect them to the cloud database.

Then I made record actions (create, update, delete) in customer record type and booking record type.

Then I made the logics in expression rules and query entities. Query entity is used to fetch the data from the database and use them in logic. Expression rules were used to write the logics of CRUD operations.

Then I went to the front. I used 8 interfaces:

1 create customer

2 update customer

3 delete customer

4 create booking

5 update booking

6 delete booking

7 view customer

8 view booking

In view customer and view booking, we have a search box, search button, filter and refresh button along with a create action which opens the interface (page) in a dialog box.

After that, we have a read only grid to read the data of customer and booking and two separate columns alongside each entry for update and delete (which also opens in a dialog box).

Working: I made 6 process models for it:

1: new booking

2: new customer

3: update booking

4: update customer

5: delete booking

6: delete customer

In these, I connected the interfaces with the database (record type) and enabled activity chaining to avoid any human approval after the interface is run.