Sri Sri University

Project - High Level Design

On

Customer Behavior Analysis

Team members:

M.S.L.V. Saranya

Aditya Narayan Panda

Sonali Nanda

Bhandari Sree Shivani

Guided by: Dr. Chinmaya Kumar Nayak

Industry Mentor:

Sumit Shukla

Table of Contents

1. Introduction 3
   1. Scope of the document 3
   2. Intended Audience 3
   3. System Overview 3
2. System Design 4-7
   1. Application Design 4-5
   2. Process Flow 5
   3. Information Flow 6
   4. Component Design 6
   5. Key Design Considerations 7
   6. API Catalogue 7
3. Data Design 8
   1. Data Model 8
   2. Data Access Mechanism 8
   3. Data Migration 8
4. Non-Functional Requirements 9
   1. Performance Aspects 9
5. References 10

**Introduction**

* 1. Scope of the document

This document contains Low Level Design of the project Customer Behavior Analysis. It also covers the data design of the dataset being used for making the dashboards.

* 1. Intended Audience

The document is intended for team members of the project, faculty supervisor and industry mentor for the purpose of tracking and reviewing the progress.

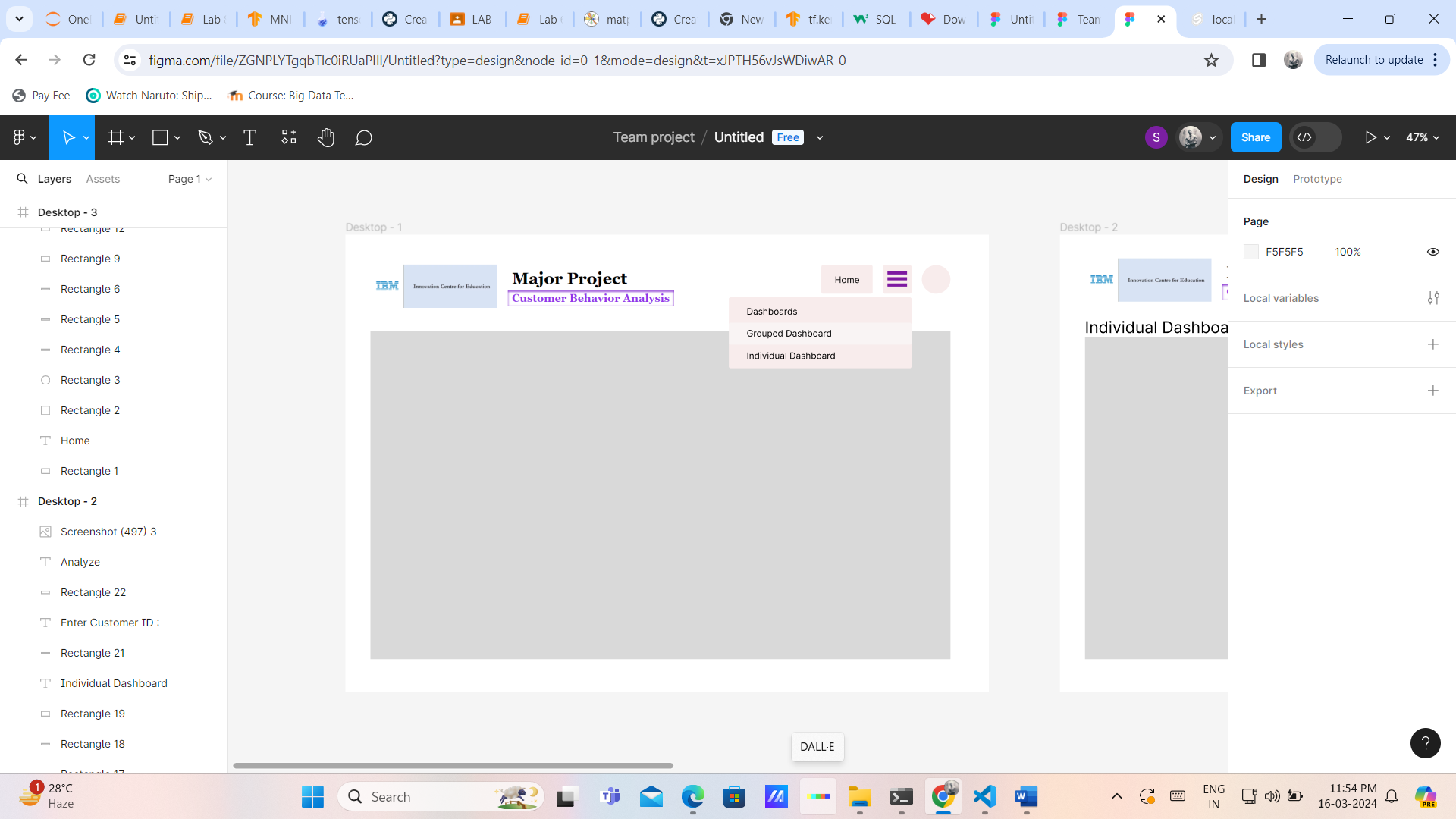
* 1. System Overview

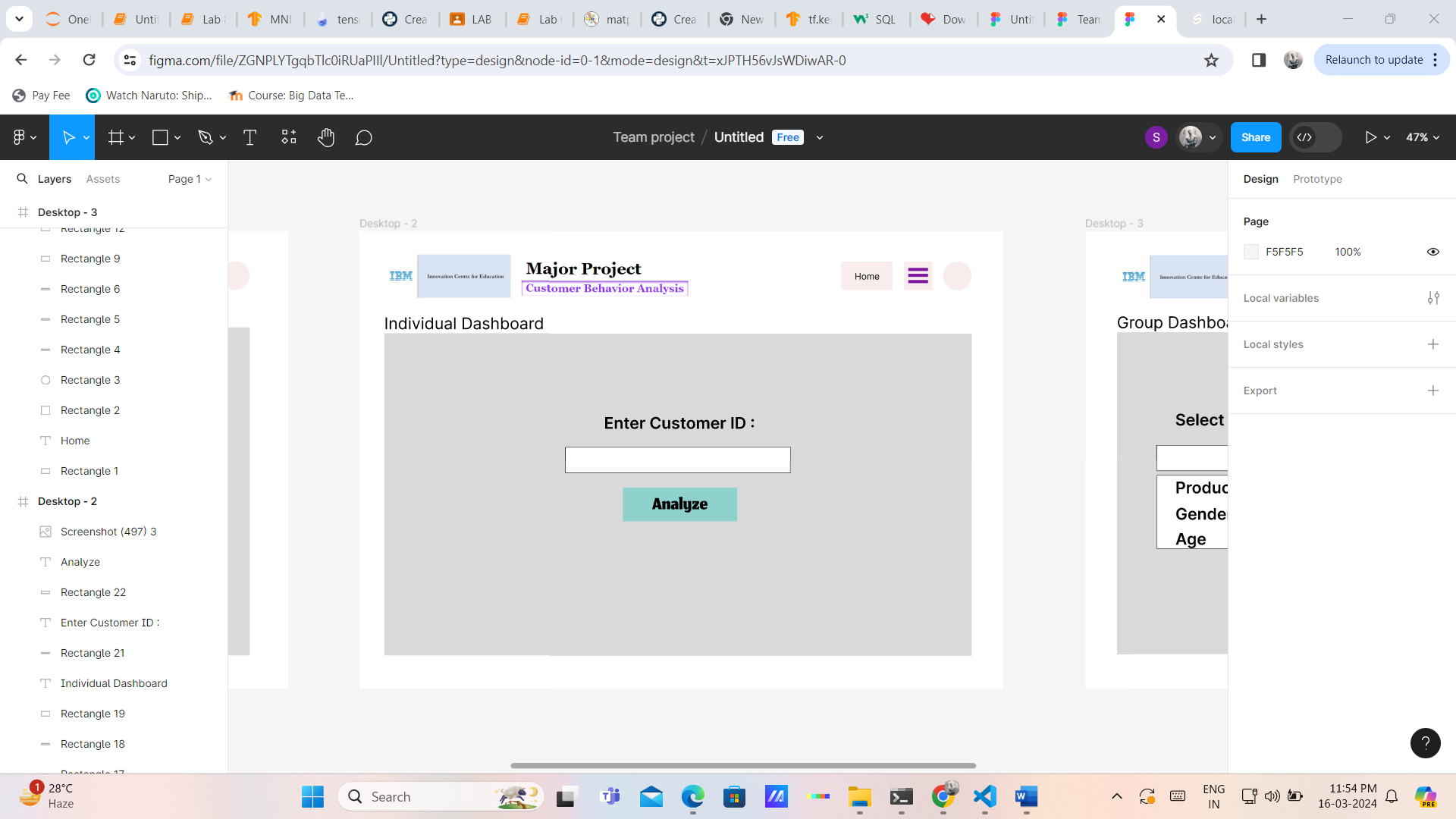
Customer Behavior Analysis is a project intended to help businesses understand their customers better. It creates dashboards based on customer data which gives valuable insights in the form of graphical method for easier understanding. The project comprises of a frontend made with SvelteKIT, backend using Django, and sqlite3 database. The dataset used for the project consists 5 lakh customer data points which will be used to create two types of dashboards:

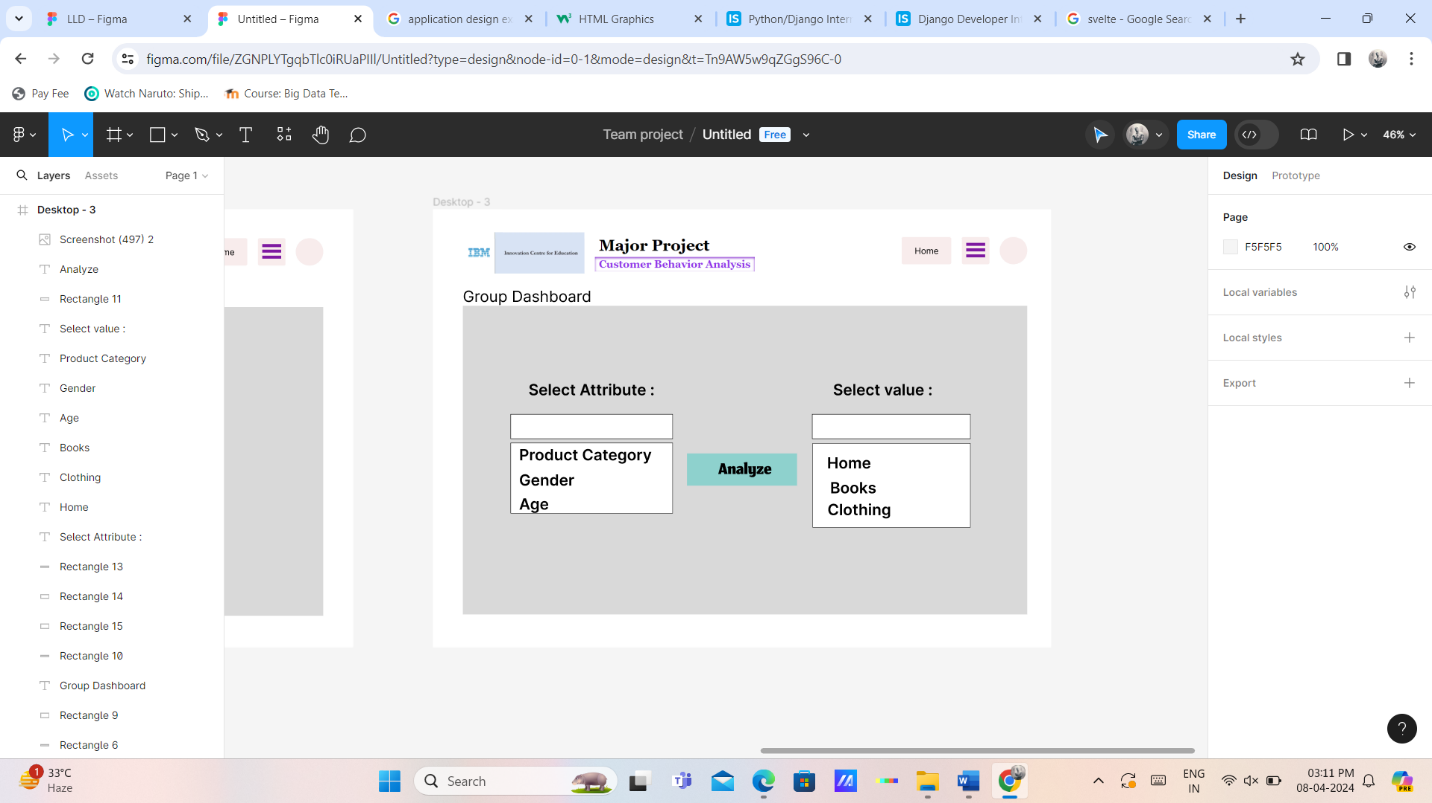
1. Individual Dashboard: Dashboard for individual customers
2. Grouped Dashboard: Dashboard based on any attribute and its value

**System Design**

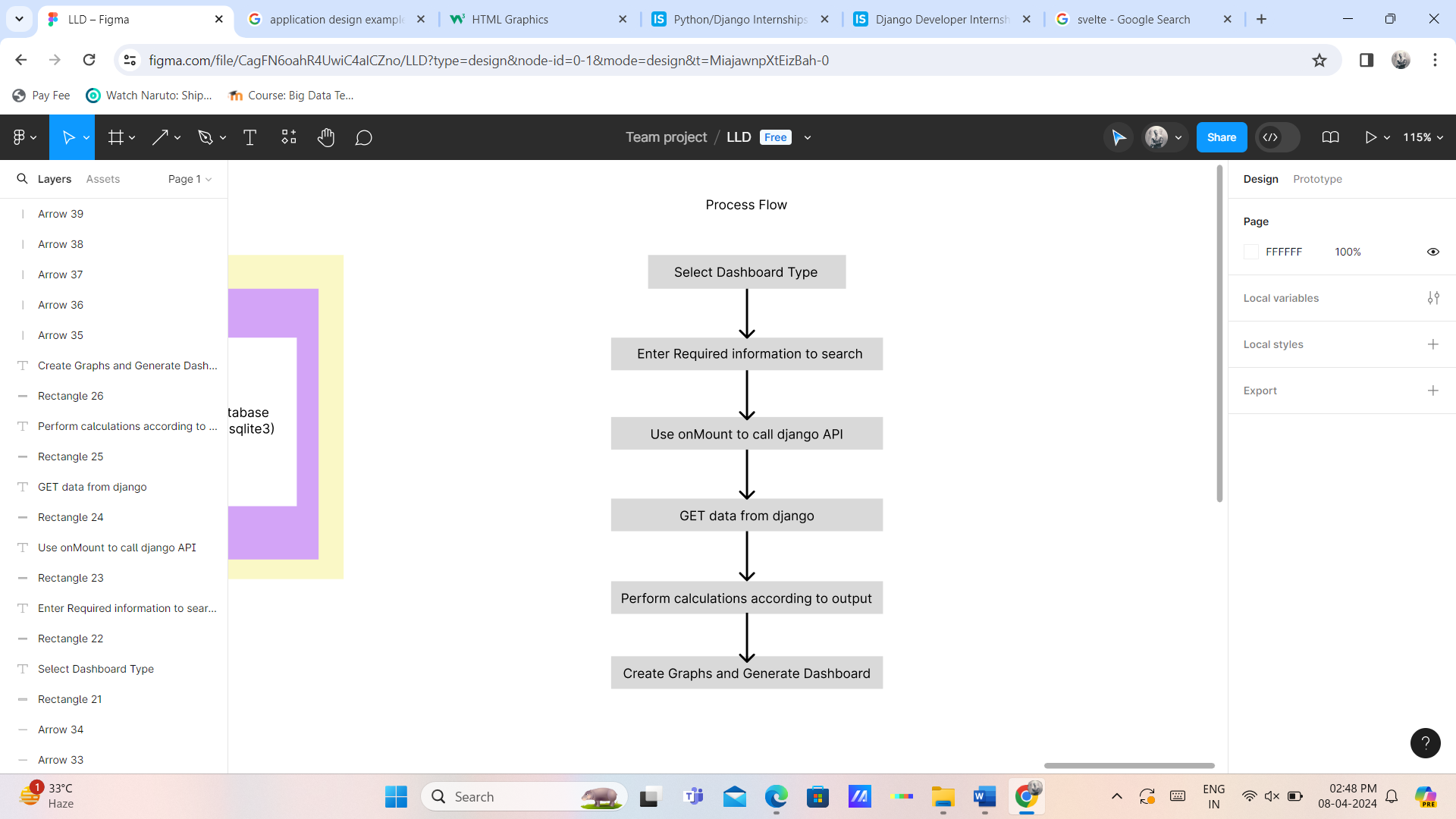
**2.1. Application Design**

****

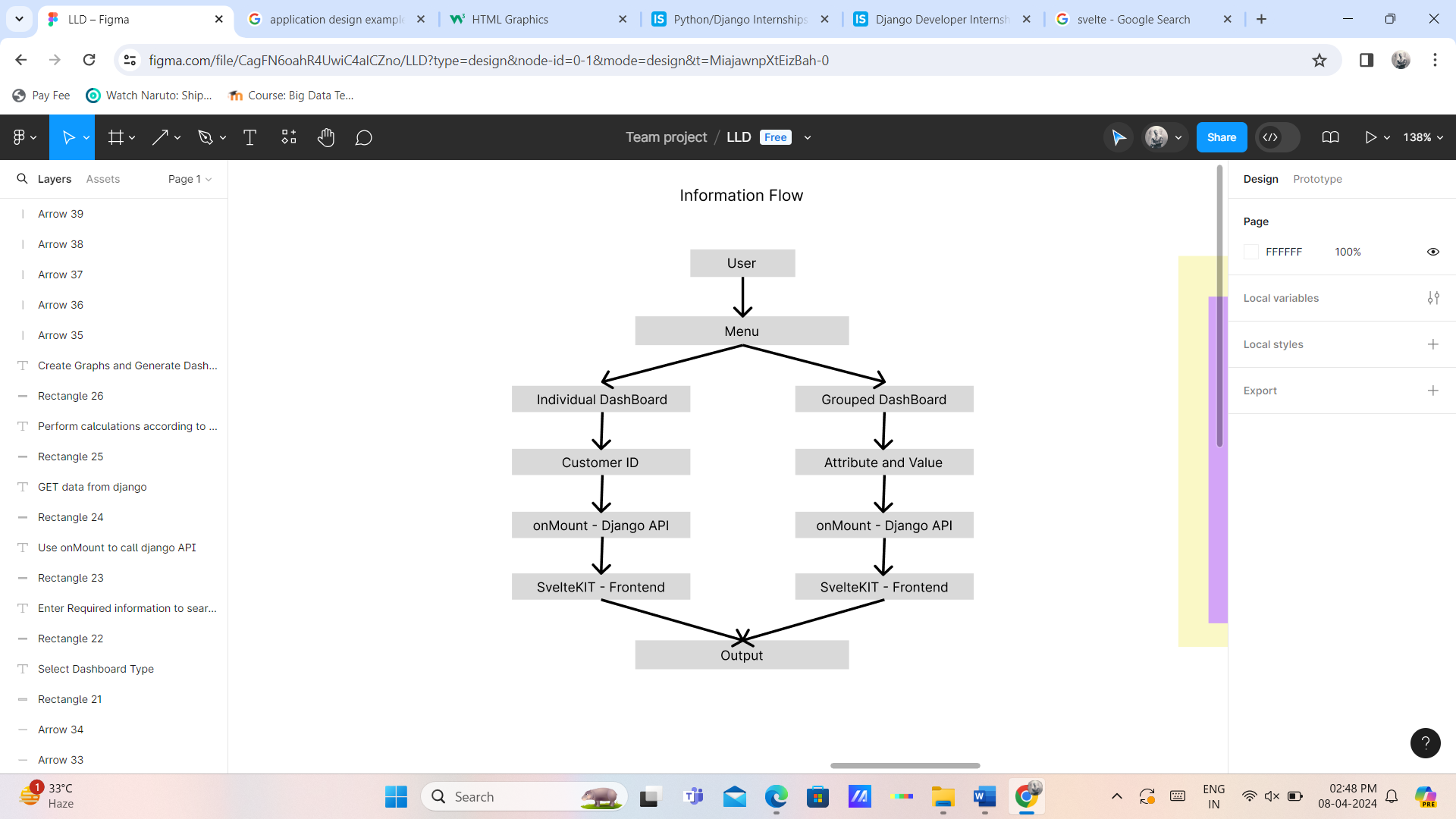
****

****

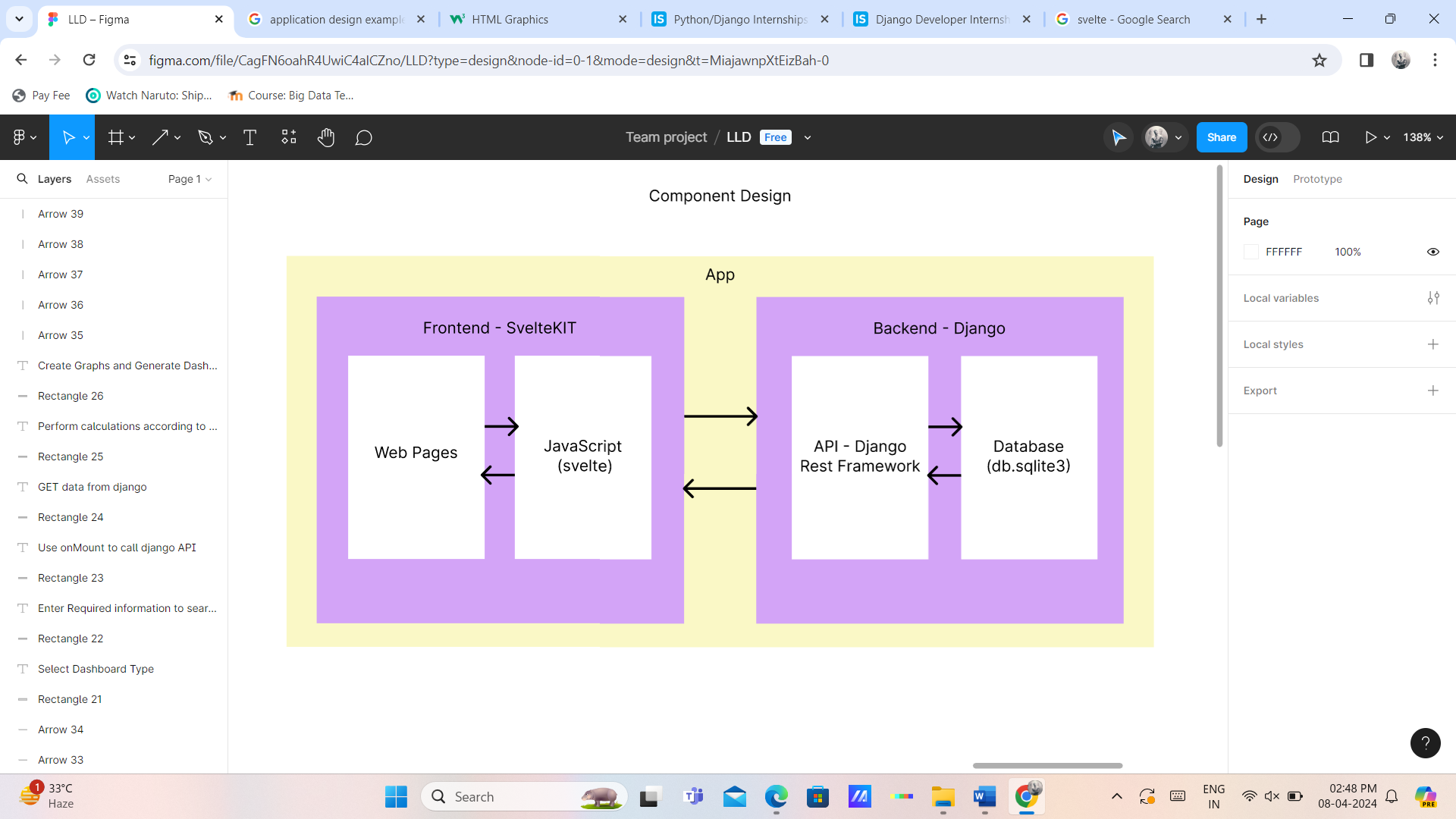
**2.2. Process Flow**

****

**2.3. Information Flow**

****

**2.4. Component Design**

****

**2.5. Key Design Considerations**

Consideration was given to following aspects in the design phase:

* The data was in csv being difficult to retrieve, so sqlite3 database was added
* To access sqlite3 database, Django backend was decided upon
* The dashboard is dynamic based on the data, so SvelteKIT was used

**2.6. API Catalogue**

The API was designed using Django Rest Framework’s ListAPIView.

11 APIs were created to get data in the json format:

/api/customers/[id]/

/api/customers/ProductCategory/[Category]/

/api/customers/PaymentMethod/[Payment]/

/api/customers/Gender/[Gender]/

/api/customers/Age/[startAge]-[endAge]/

/api/customers/P\_Date/Month/[Month]/

/api/customers/P\_Date/Year/[Year]/

/api/customers/P\_Date/Date/[Date]/

/api/customers/TPAmt/[startTPAmt]-[endTPAmt]/

/api/customers/Returns/[Returns]

/api/customers/Churn/[Churn]

**Data Design**

**3.1. Data Model – Customer (db.sqlite3)**

Cid=models.IntegerField()

P\_Date=models.DateTimeField()

Category=models.CharField(max\_length=64)

Price=models.FloatField()

Quantity=models.IntegerField()

TPAmt=models.FloatField()

Payment=models.CharField(max\_length=64)

Returns=models.FloatField()

Name=models.CharField(max\_length=128)

Age=models.IntegerField()

Gender=models.CharField(max\_length=8)

Churn=models.IntegerField()

**3.2. Data Access Mechanism**

The data is accessed through various APIs created in the Django Rest Framework.

**3.3. Data Migrations**

The data was originally in csv format and was divided into two files.

Later it was migrated to a single model “Customer” inside the database db.sqlite3 within the backend.

**Non-Functional Requirements**

**4.1. Performance Aspects**

Data size : 500000 x 12

Required RAM : 6GB or higher

Required Processor : Intel Core i3 10th GEN alternative or higher

Both backend and frontend should be running for successfully implementation.

**References**

* <https://kit.svelte.dev/docs/introduction>
* <https://docs.djangoproject.com/en/5.0/>
* <https://www.django-rest-framework.org/>