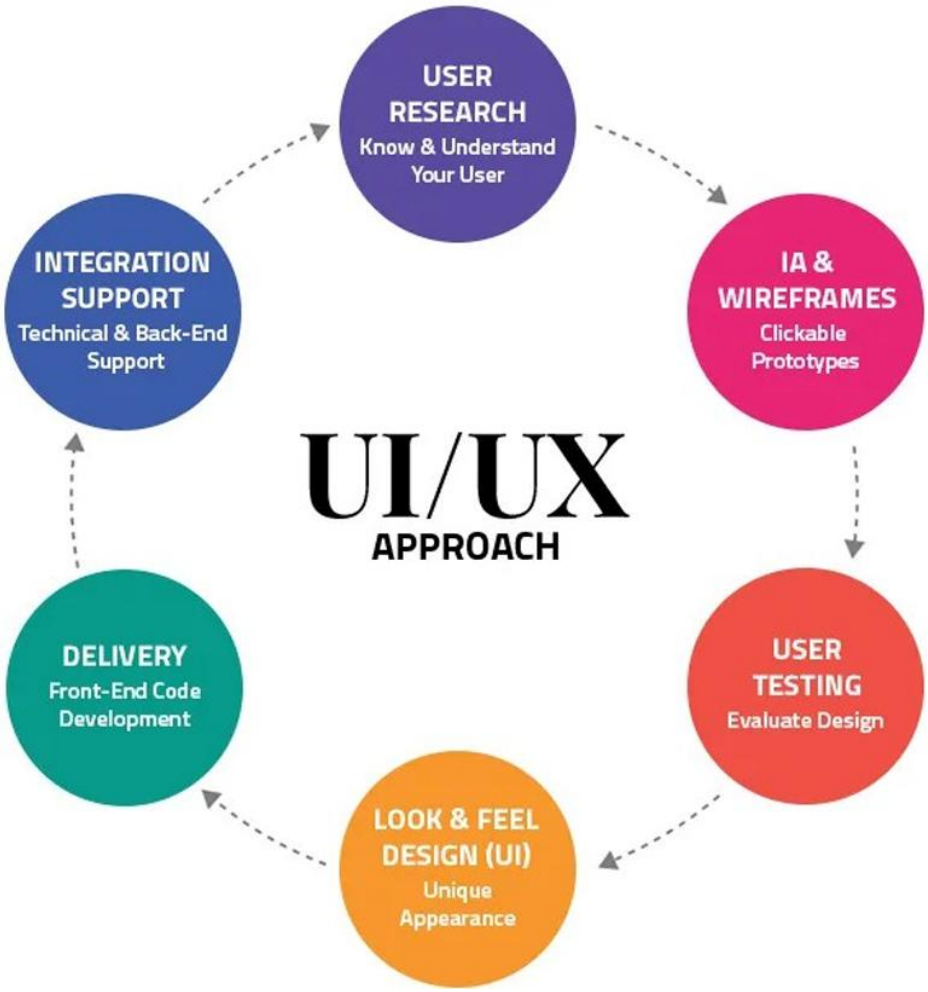
Experiment No. 1

Introduction to UI Lifecycle and UI Tools

UI Lifecycle:



UI Tools:

InVision: InVision is a web-based prototyping tool popular with both UX and UI designers alike. You can upload static design files and quickly turn them into high-fidelity, interactive prototypes.

Sketch: Sketch is a vector-based digital design app that every UI pro needs. You can easily resize anything you draw without losing sharpness.

Figma: Figma is the first in-browser interface design tool. With powerful editing tools and many useful features, Figma is a one-stop shop for designing, prototyping, and gather ing feedback. UI designers can take advantage of the constraints feature, which adapts designs when the screen size changes. The components feature also makes it easy to reuse elements across designs.

Flinto: Flinto is an interactive prototyping app for Mac. It offers everything you need to bring your designs to life, including features for designing micro-interactions, screen transitions, video layers, UI sound effects, and customizable scrolling.

Adobe XD: Adobe XD is a vector-based tool for designing and prototyping user experi encesfor web, mobile, andevenvoiceinterfaces. If you're familiar with the Adobe Creative Cloudsuite, you'll feel right at home in Adobe XD.It's an extremely versatile tool for design ing, prototyping, sharing, collaborating, and creating a complete design system. Adobe XD supports Windows 10, macOS, and has mobile apps for both Android and iOS.

Experiment No. 2

Project Proposal and Requirement Gathering (Introduction of the Project)

1. Project Title:

SastaTours – A Budget-Friendly Tour Planning Mobile Application

2. Introduction:

Traveling is a dream for many, but high expenses often become a barrier, especially for students, solo travelers, and middle-income groups. To address this, **CheapTrip** is proposed as a budget tour planning application designed specifically for users who wish to travel affordably without compromising basic comfort and experience.

This application will allow users to explore **low-cost tour packages, local guides, budget accommodations, affordable transport options**, and community travel experiences. Unlike premium tour apps focusing on luxury and premium deals, CheapTrip will focus on **cost efficiency, local exploration, and user-generated tips** to help others save money during their trips.

3. Objective of the Project:

The primary objective is to **develop an application that enables cost-conscious users to plan and execute travel at the lowest possible expense** by aggregating affordable resources and providing real-time, user-based recommendations.

4. Target Users:

- Budget travelers
- College students
- Backpackers
- Low-income families
- Adventure seekers looking for local experiences

5. Scope of the Project:

The CheapTrip app will:

- Provide listings of budget hotels, hostels, and homestays.
- Show low-cost travel routes (bus, local trains, shared cabs).
- Offer curated travel plans under a fixed budget (e.g., "2 Days in Goa under ₹1000").
- Include a community section for travelers to share tips, hacks, and low-cost destinations.

- Enable cost comparison across various services.
- Offer language support and offline maps for remote areas.

6. Key Features (Initial Requirements):

- User registration and login system
- Budget travel packages with filter options
- Integration with public transport schedules
- Maps and directions to low-cost attractions
- Review and rating system for places and services
- “Suggest a Trip” tool based on budget and interest
- Local guides contact information and ratings
- User-generated content (travel stories, vlogs, tips)

7. Tools & Technologies (Tentative):

- **Frontend:** React Native / Flutter (for cross-platform support)
- **Backend:** Node.js or PHP (Laravel for lightweight frameworks)
- **Database:** MySQL or Firebase
- **APIs:** Google Maps API, public transport APIs
- **Hosting:** Firebase Hosting or cheap shared hosting solutions

8. Limitations:

- Limited to only budget options (not for premium users)
- Dependent on user-generated content for updates
- May not offer live booking; rather, it provides redirection to low-cost services

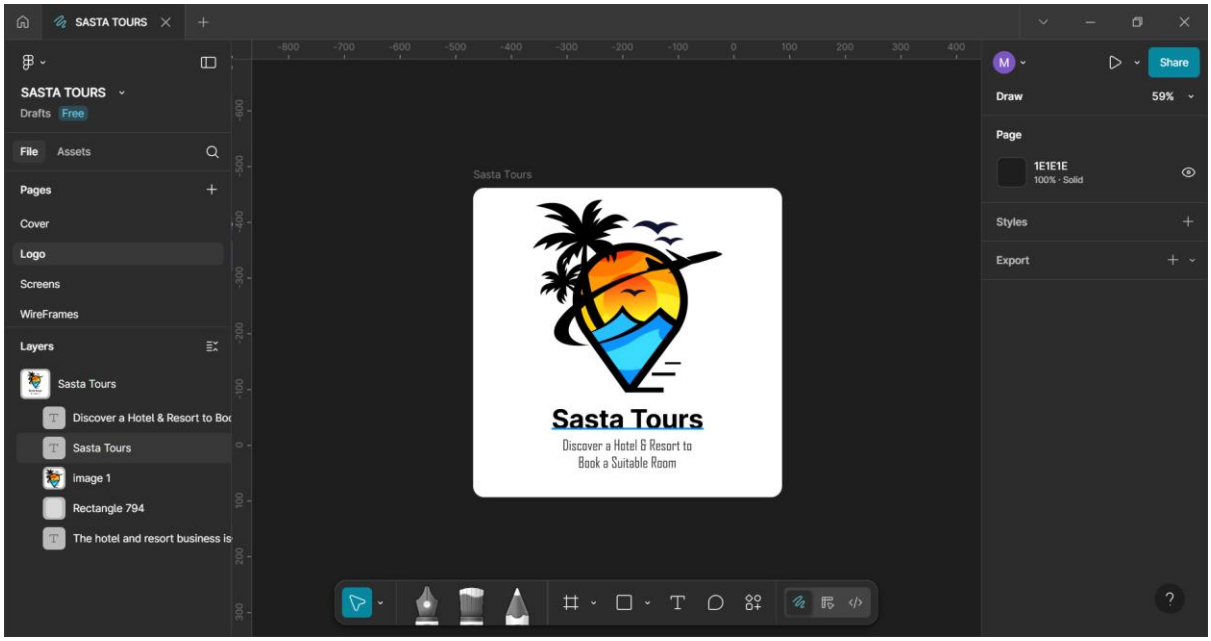
9. Conclusion:

The CheapTrip application aims to **democratize travel by making it accessible to everyone**, especially those who cannot afford expensive trips. By focusing on community-based content, budget optimization, and local resource mapping, this app has the potential to fill a gap in the current travel app ecosystem.

Experiment No. 3

Logo Designing

Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1



Experiment No. 4

Problem Statement:

Despite the growing popularity of travel among young people, students, and low-income individuals, most existing travel applications primarily cater to premium customers and offer costly packages, accommodations, and transport services. As a result, **budget-conscious travelers struggle to find reliable and consolidated information on affordable travel options**. They often rely on scattered forums, outdated blogs, or word of mouth to plan trips, leading to **confusion, missed opportunities, or overspending**.

There is a significant gap in the market for a centralized platform that **curates and suggests cheap and verified travel resources**, including low-cost accommodations, public transport routes, affordable destinations, and community-generated tips for cost-saving. Travelers need a **simple, lightweight, and user-friendly application** that helps them plan enjoyable trips within a tight budget.

System Concept Statement:

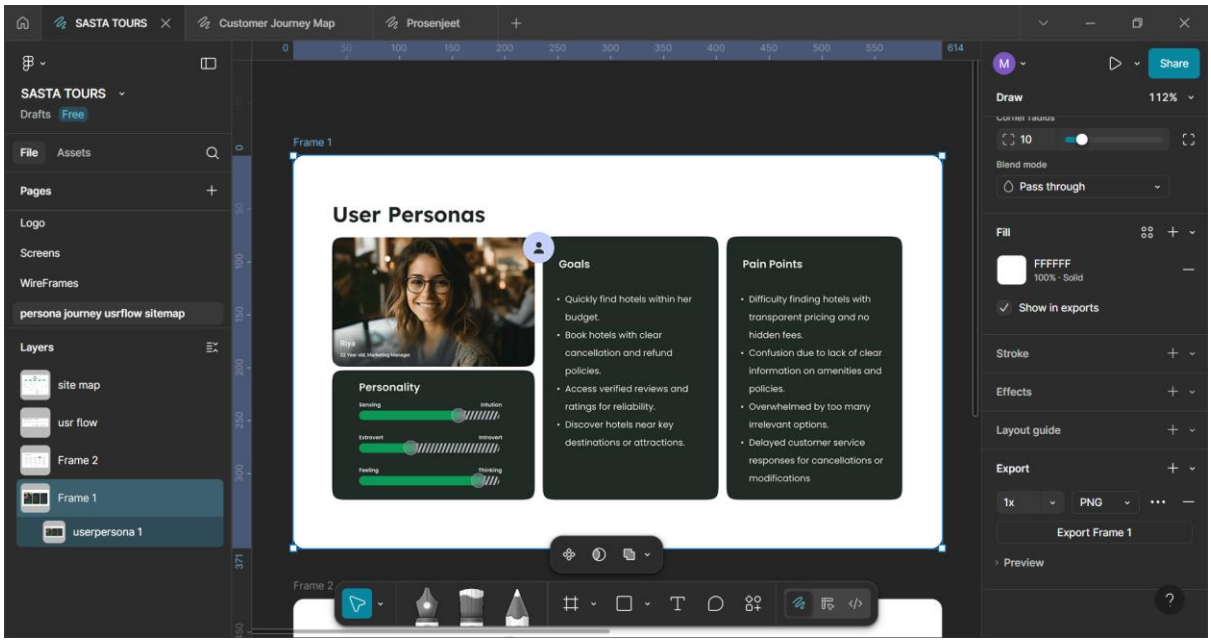
The proposed system, **CheapTrip**, is a mobile-based tour planning application that provides a **centralized platform for budget travel planning**. It will allow users to discover and organize trips using only **low-cost resources** such as public transport, hostels, homestays, and cheap local attractions. The system will support **search filters** based on destination, travel dates, and budget limits to suggest optimized travel plans.

Users will also be able to share travel tips, reviews, itineraries, and hidden gems, helping others travel smarter. The application will offer a **community-driven experience**, integrating budget travel features, real-time suggestions, offline maps, and low-cost service listings, making travel more accessible and affordable for all.

Experiment No. 5

Design a User Persona

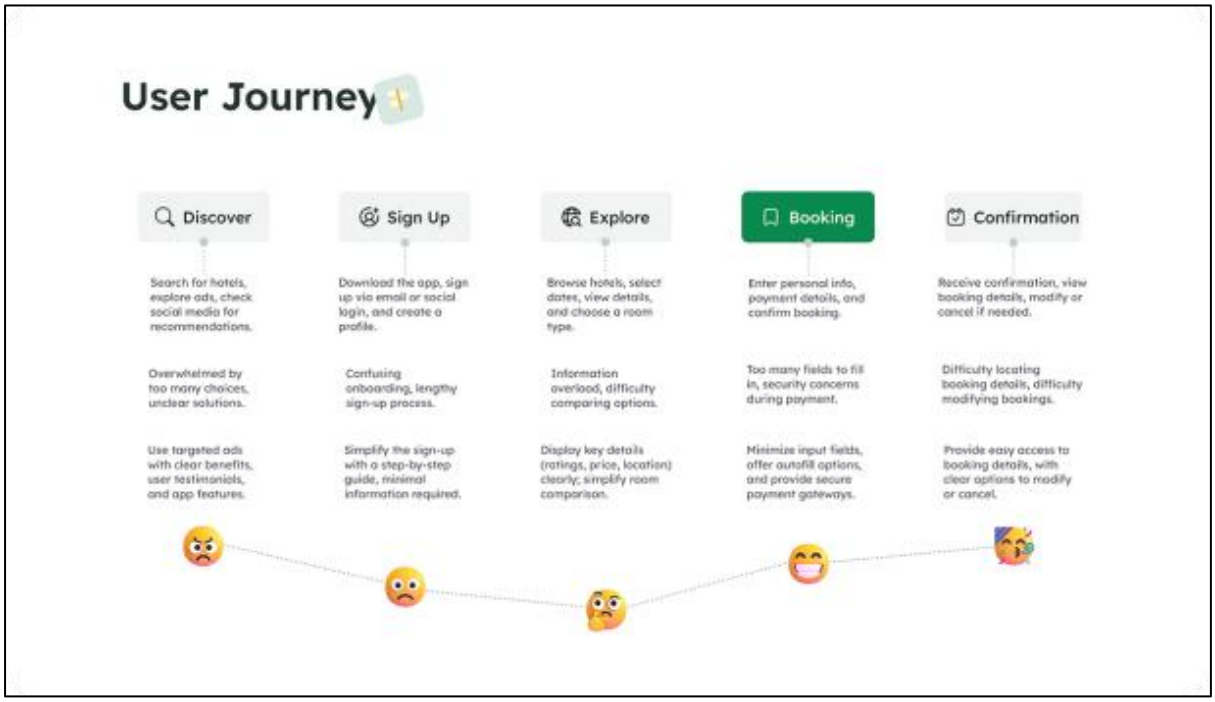
Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1

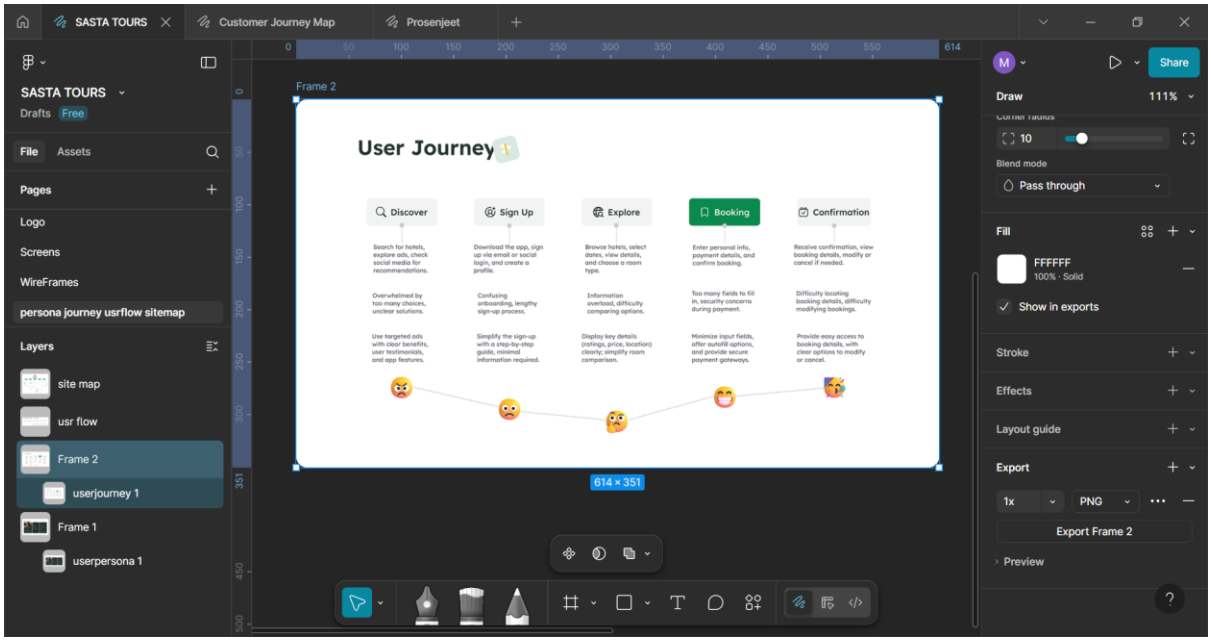


Experiment No. 6

Design a Customer Journey Map

Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1





Experiment No. 7

Tour Management System – ER Diagram Overview

1. What is an ER Diagram?

An **Entity Relationship (ER) Diagram** is a visual representation of the data and relationships in a system. It is primarily used for database design to model how entities such as users, cars, and bookings interact.

2. Purpose in the Tour Management System

The ER Diagram of the **Tour Management System** illustrates the interaction between:

- Users
- Rental Companies
- Cars
- Bookings
- Payments

It serves as a blueprint for implementing the system's backend database.

Project Introduction: Tour Management System

The **Tour Management System** is designed to meet the increasing demand for convenient and flexible vehicle rental options.

Goals:

- Offer a **cost-effective** alternative to owning a vehicle.
- Enable users to select cars based on **budget**, **type**, and **rental duration**.
- Allow rental companies to **manage their fleet and bookings**.

Objectives

1. Provide a user-friendly car rental experience.
2. Reduce private car ownership and urban traffic congestion.
3. Improve vehicle accessibility for individuals without personal cars.
4. Support rental businesses with a streamlined booking and management platform.

ER Model Notations and Symbols

Symbol	Description
Rectangle	**Entity** (e.g., User, Car, Booking)
Ellipse	**Attribute** (e.g., Name, Car_Type)
Diamond	**Relationship** (e.g., Books, Owns, Pays)
Line	Connects attributes to entities or entities to relationships

Entities and Their Attributes

1. User

- *User_ID* (Primary Key)
- Name
- Email
- Phone_Number

2. Car

- *Car_ID* (Primary Key)
- Car_Type
- Model
- Rent_Per_Day
- Availability_Status

3. Rental_Company

- *Company_ID* (Primary Key)
- Name
- Location
- Contact_Details

4. Booking

- *Booking_ID* (Primary Key)
- Booking_Date
- Rent_Duration
- Status

5. Payment

- *Payment_ID* (Primary Key)
- Amount
- Payment_Date
- Payment_Method

Relationships

1. User – Books – Booking

- A **User** can book multiple **Bookings**
- Each **Booking** is associated with one **User**

2. Booking – Includes – Car

- Each **Booking** involves one **Car**
- A **Car** can be involved in multiple **Bookings**

3. Rental_Company – Owns – Car

- A **Rental_Company** owns many **Cars**
- Each **Car** is owned by one **Rental_Company**

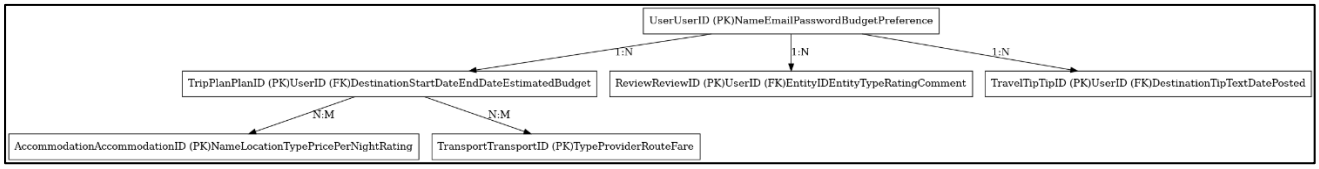
4. User – Makes – Payment

- A **User** makes one or more **Payments**
- Each **Payment** is made by a **User**

5. Booking – Generates – Payment

- Each **Booking** results in one **Payment**
- A **Payment** is associated with one **Booking**

ER Diagram (Text Description Layout)



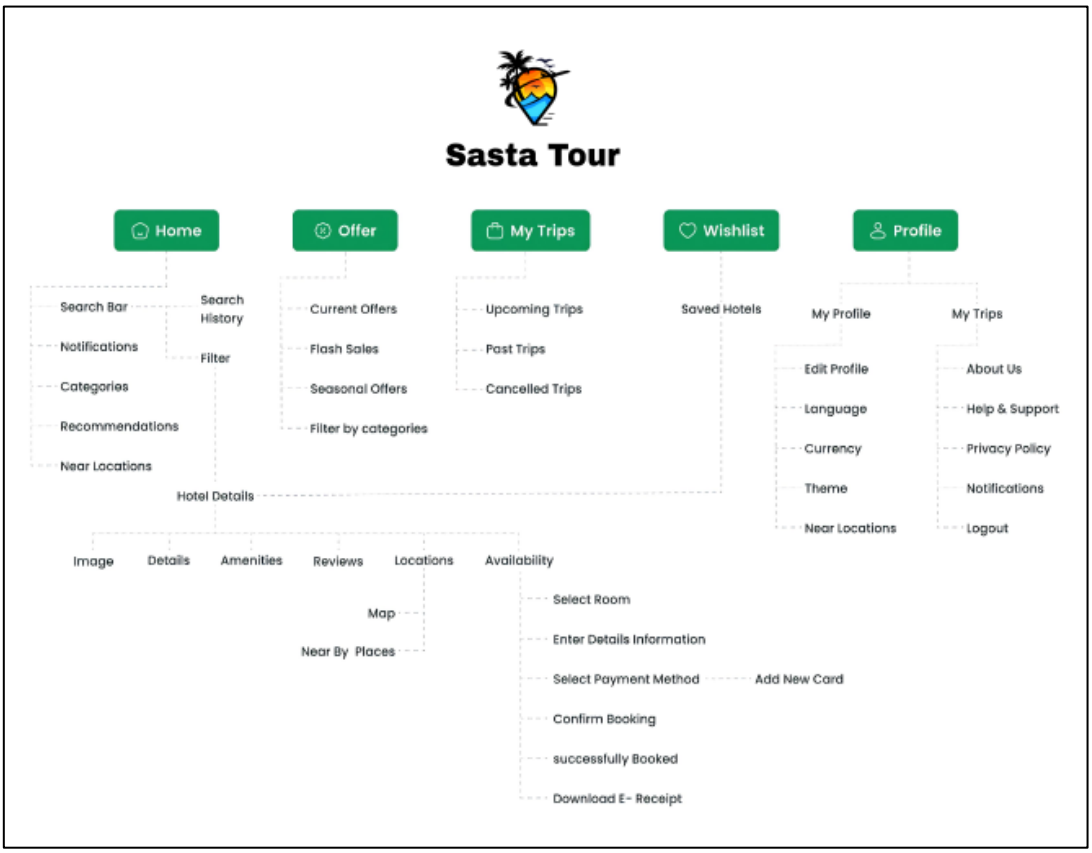
Experiment No. 8

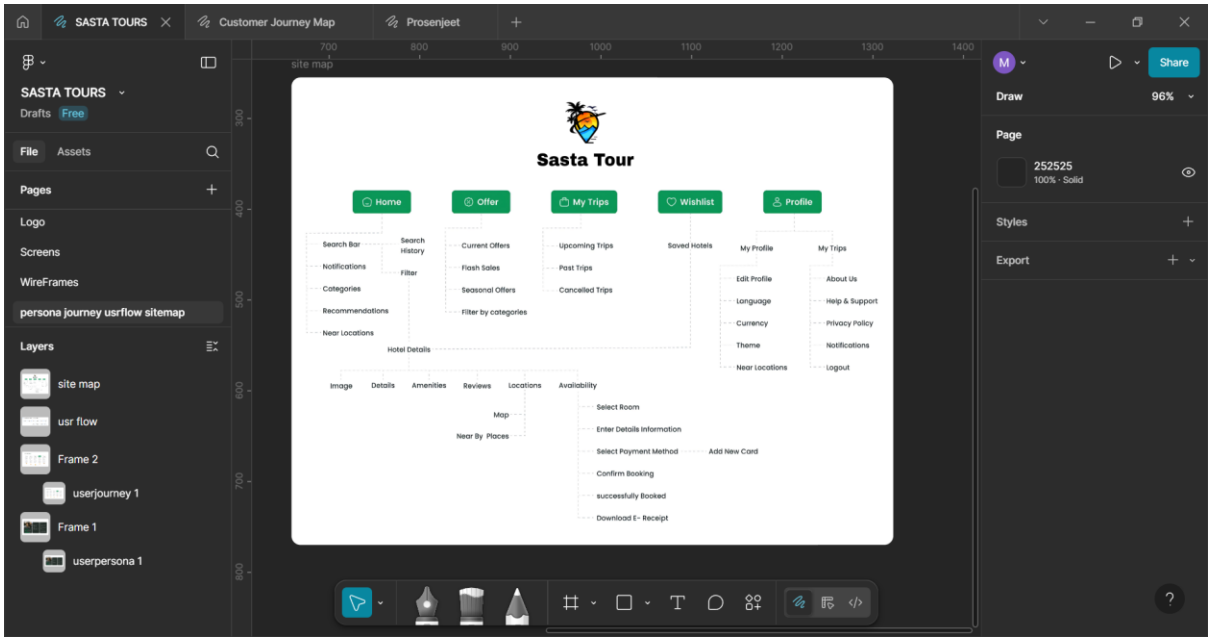
Creation of scenario – Story Board

Experiment No. 9

Create a Site Map

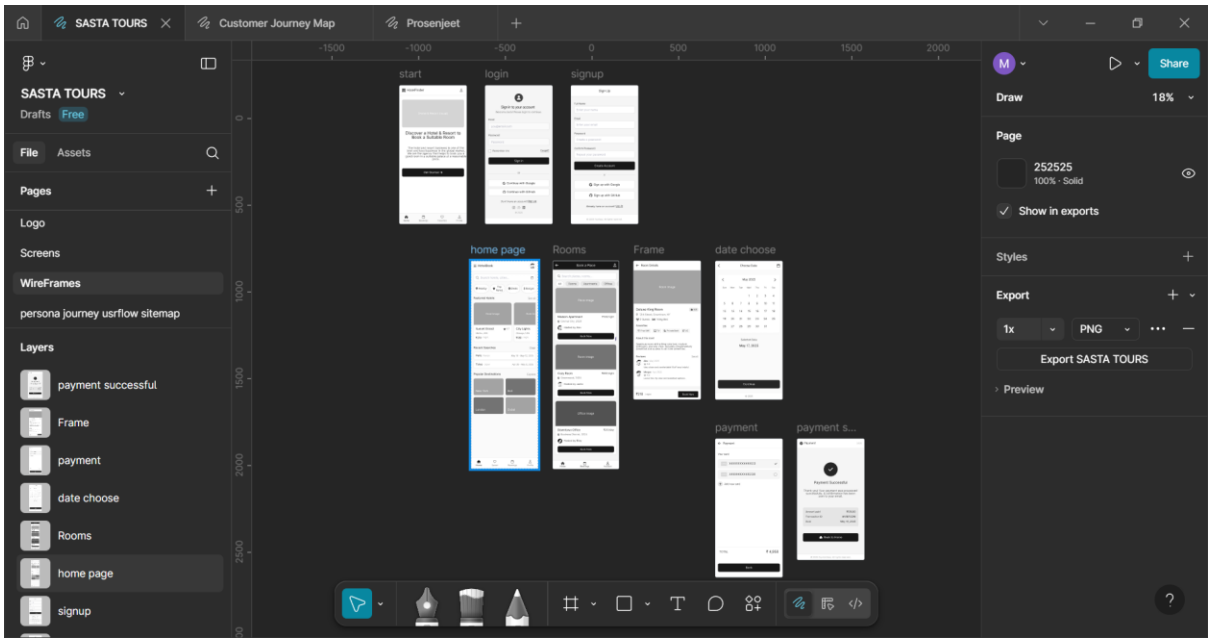
Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1





Experiment No. 10

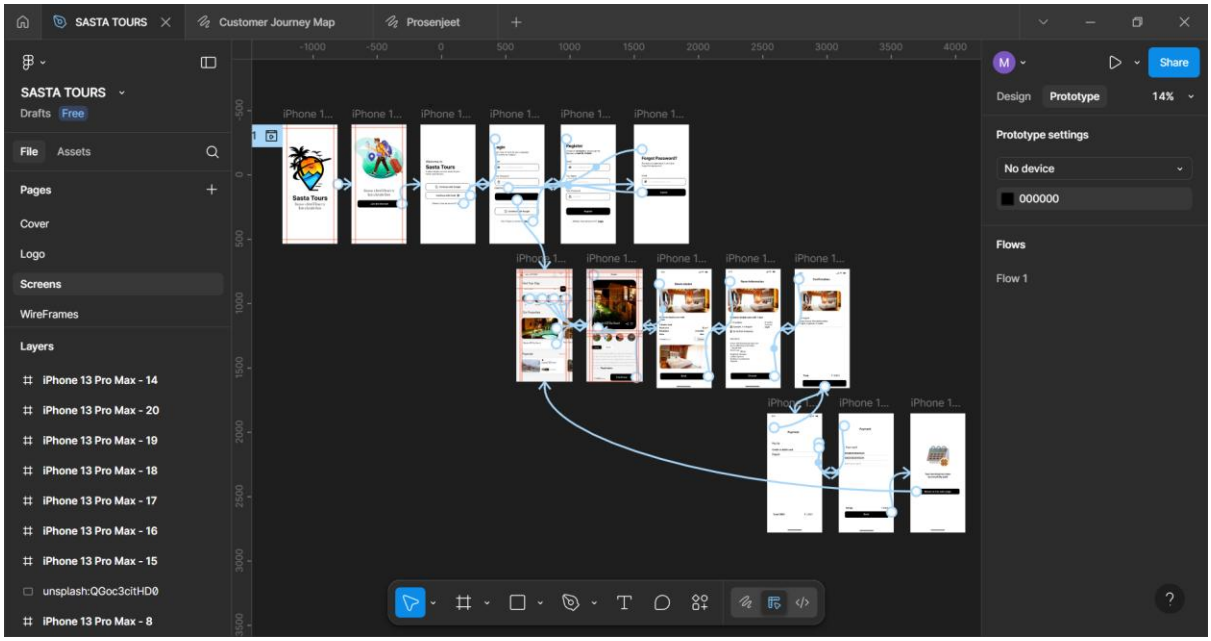
Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1



Experiment No. 11

Create Prototype for Chosen Project

Link: https://www.figma.com/design/MHQvWKNkYgKsxWygkddZN8/SASTA-TOURS?m=auto&t=6L0bXIV3dSOJFcj4-1



Experiment No. 12

UX Evaluation of Chosen Project

Project: Tour Management System

User Experience (UX) Evaluation

The user interface of the **Tour Management System** was evaluated by third-party users to ensure usability, simplicity, and efficiency. Evaluation was done based on:

- **Ease of Navigation**
- **Visual Design & Layout**
- **Responsiveness and Load Time**
- **User Feedback Mechanism**
- **Accessibility of Key Features**

Third-Party Testing

Test scripts were given to external users to perform standard user flows such as:

- Searching for available cars
- Booking a vehicle
- Making a payment
- Cancelling a booking

Feedback from testers helped identify minor usability issues, which were addressed in the final version.

Software Testing

Testing ensures that the software performs correctly and is free from errors. Proper testing leads to more reliable and robust applications.

Types of Testing

1. Unit Testing

- **Purpose:** Tests the smallest parts of the program (individual units or functions).
- **Performed By:** Programmers.
- **Process:** Sample inputs are given to check if the output is as expected.

2. Integration Testing

- **Purpose:** Tests the interaction between integrated modules.
- **Goal:** To ensure that unit-tested components work together correctly.

3. Regression Testing

- **Purpose:** Ensures that newly added code does not break existing functionality.
- **Use Case:** After updating or adding new features/modules.

4. Smoke Testing

- **Purpose:** Verifies whether the software build is stable enough for further testing.
- **Also Known As:** "Build Verification Testing."

5. Alpha Testing

- **Type:** Acceptance Testing.
- **Performed By:** Internal QA team.
- **When:** Before releasing the product to customers.

6. Beta Testing

- **Performed By:** Actual users at customer sites.
- **Purpose:** Real-time feedback in actual usage environment.

7. System Testing

- **Scope:** Tests the complete system on various platforms and operating systems.
- **Type:** Black-box testing.
- **Focus:** Input/output behavior rather than internal code.

8. Stress Testing

- **Purpose:** Evaluates software performance under extreme conditions.
- **Goal:** To determine software limits and ensure it doesn't crash.

9. Performance Testing

- **Focus:** Measures software response time, speed, scalability, and resource usage.
- **Goal:** Ensure optimal performance under load.

10. Acceptance Testing

- **Performed By:** End users or customers.
- **Purpose:** To verify if the system meets all business requirements and functions as expected.

No	Action	Input	Expected Output	Actual Output	Test Result	Test Comment
1	Launch Application	Click on software	Login Page should appear	Login Page appeared	Pass	Application started successfully
2	Enter correct login	Username: user1, Password: ****	Redirect to Homepage	Redirected to Homepage	Pass	Login functionality working
3	Wrong credentials	Username: wrong, Password: wrong	Display "Login Failed"	Displayed "Login Failed"	Pass	Error correctly shown on wrong credentials
4	Leave Email blank	Leave field empty	Display "Email required"	Displayed "Email required"	Pass	Field validation works as expected
5	Invalid Email format	Email: abc.com	Display "Invalid Email"	Displayed "Invalid Email"	Pass	Email format validation successful
6	Valid Email	Email: user@example.com	No error message	No error message displayed	Pass	Valid email accepted
7	Name with numbers	Name: 12345	Display "Invalid Name"	Displayed "Invalid Name"	Pass	Name field validation is functional
8	Valid name	Name: John Doe	No error message	No error message displayed	Pass	Valid name accepted successfully
9	Phone with letters	Phone: abcdefg	Display "Invalid Number"	Displayed "Invalid Number"	Pass	Non-numeric phone validation works
10	Valid phone	Phone: 9876543210	No error message	No error message displayed	Pass	Valid phone number accepted
11	Empty Sign Up	Leave all fields blank	Show warning for each field	Warnings displayed for all fields	Pass	Comprehensive validation functioning
12	One field blank	Leave phone field empty	Show "Phone required"	Displayed "Phone required"	Pass	Field-specific error shown
13	Email already exists	Email: user1@example.com	Display "Email already registered"	Displayed "Email already registered"	Pass	Duplicate email correctly handled