



Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY

University with Graded Autonomy Status

(An ISO 21001 : 2018 Certified Institution)

Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.



RECORD NOTEBOOK

USER EXPERIENCE DESIGN
- (EBCS22ET5)

2024-2025 (ODD SEMESTER)

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

NAME : KUMARAN T

REGISTER NO : 231061201005

COURSE : B.TECH CSE (P.T)

YEAR/SEM/SEC : II/ III



Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY

University with Graded Autonomy Status

(An ISO 21001 : 2018 Certified Institution)

Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.



BONAFIDE CERTIFICATE

Register No: 231061201005

Name of Lab : USER EXPERIENCE DESIGN – (EBCS22ET5)

Department : COMPUTER SCIENCE AND ENGINEERING

Certified that this is Bonafide record of work done by KUMARAN T (231061201005) of **II** Year B-Tech CSE in the **USER EXPERIENCE DESIGN** during the year 2024-2025.

Signature of Lab-in-Charge

Signature of Head of Dept

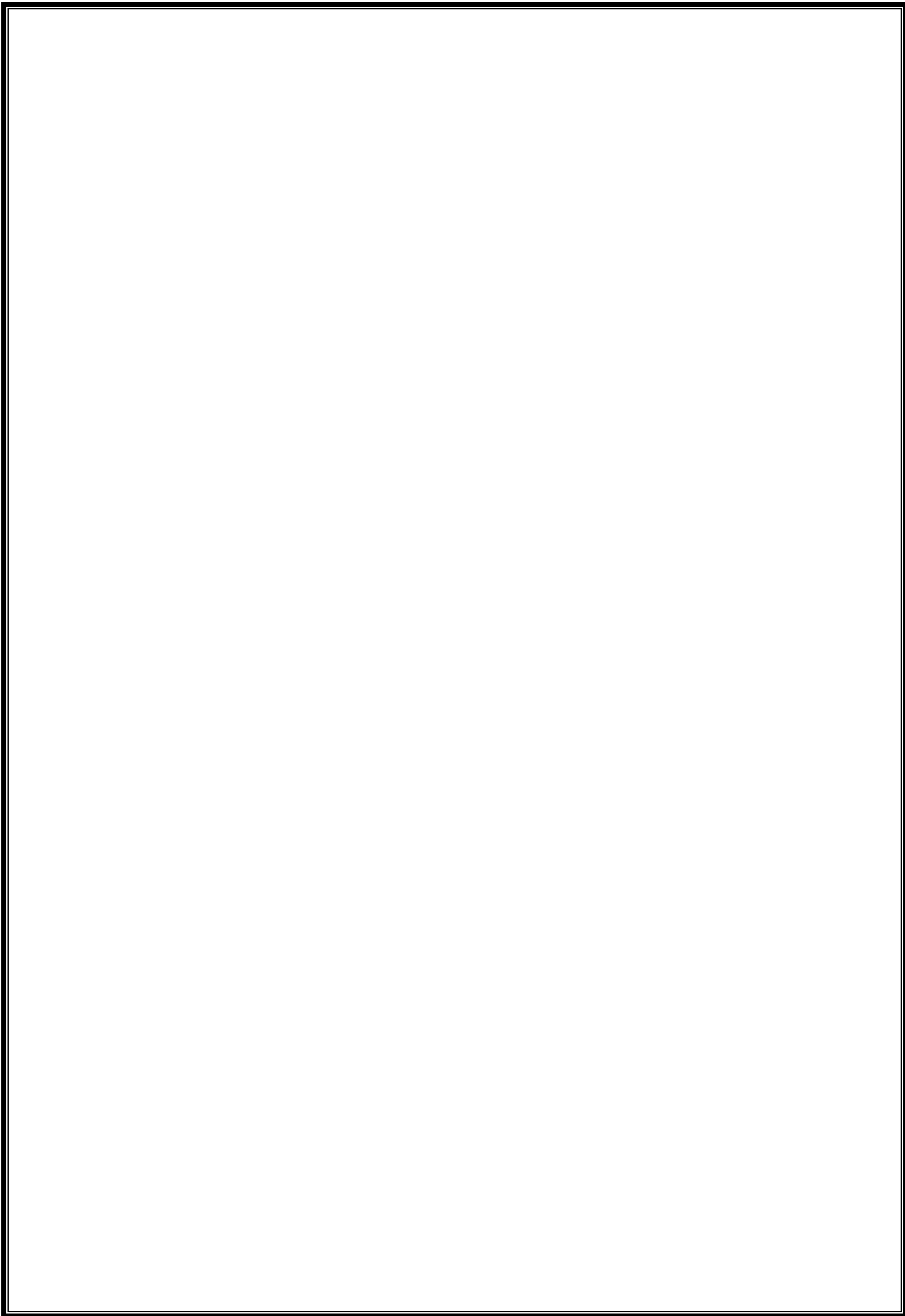
Submitted for the Practical Examination held on _____

Internal Examiner

External Examiner

CONTENTS

S.NO	DATE	NAME OF THE EXPERIMENT	PAGE NO	STAFF SIGNATURE
1		Designing a responsive layout for a societal application	1	
2		Exploring various UI Interaction Patterns	7	
3		Developing wireflow diagram for application using open-source software	10	
4		Hands on Design Thinking Process for a new product	12	
5		Brainstorming feature for proposed product	15	
6		Identify a customer problem to solve	19	
7		Conduct end-to-end user research - User research, creating personas, Ideation process (User stories, Scenarios), Flow diagrams, Flow Mapping.	23	
8		Sketch, design with popular tool and build a prototype and perform usability testing and identify improvements	27	



EX.01: Designing a responsive layout for a societal application

Aim: To design a responsive layout for a societal application.

Algorithm:

1. HTML and CSS Setup:

- Create an HTML5 document with character encoding and viewport settings.
- Use internal CSS to style the layout components.

2. Reset Default Styles:

- Reset margins, padding, and specify a font-family for better control.

3. Style Header, Navigation, Content, and Footer:

- Apply background colors, text colors, and alignment to the header, navigation, and footer.
- Style navigation links as inline elements with spacing.
- Center-align text in header, navigation, and footer.

4. Implement Responsive Design:

- Use a media query for screens up to 768px wide.
- Adjust navigation for mobile display (block-level elements with margin).

5. Add Content:

- Place your application's content within the .container div.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Societal Application</title>
  <style>
    /* Reset some default styles */
    body, html {
      margin: 0;
      padding: 0;
      font-family: Arial, sans-serif;
    }
    /* Header styles */
    header {
      background-color: #ff0000;
      color: #fff;
      padding: 10px;
      text-align: center;
    }
    /* Navigation styles */ nav {
      background-color: #47fff0;
      color: #fff;
      text-align: center;
    }

    nav ul {
      list-style: none;
      padding: 0;
    }
    nav li {
      display: inline;
      margin: 0 15px;
    }

    /* Main content styles */
    .container {
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
    }
  /* Responsive design */
```

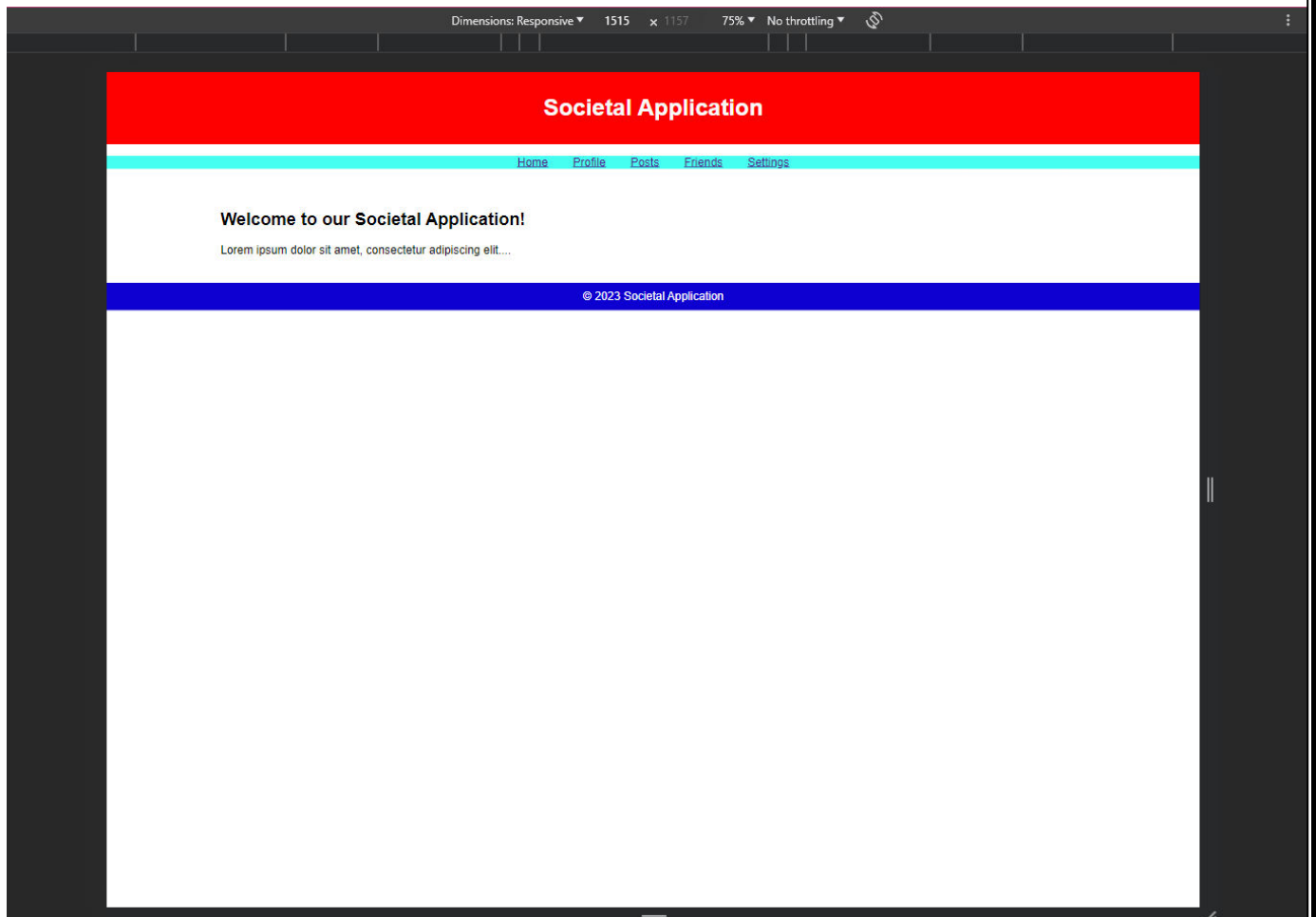
```

        @Media (max-width: 768px) {
            nav {
                display: block;
                text-align: center;
            }
            nav li {
                display: block;
                margin: 10px 0;
            }
        }
        /* Footer styles */
        footer {
            background-color: #0e00d1;
            color: #fff;
            text-align: center;
            padding: 10px;
        }
    </style>
</head>
<body>
    <header>
        <h1>Societal Application</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li><a href="#">Posts</a></li>
            <li><a href="#">Friends</a></li>
            <li><a href="#">Settings</a></li>
        </ul>
    </nav>
    <div class="container">
        <!-- Your content goes here -->
        <h2>Welcome to our Societal Application! </h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipescen
elite....</p>
    </div>
    <footer>
        &copy; 2023 Societal Application
    </footer>
</body>
</html>

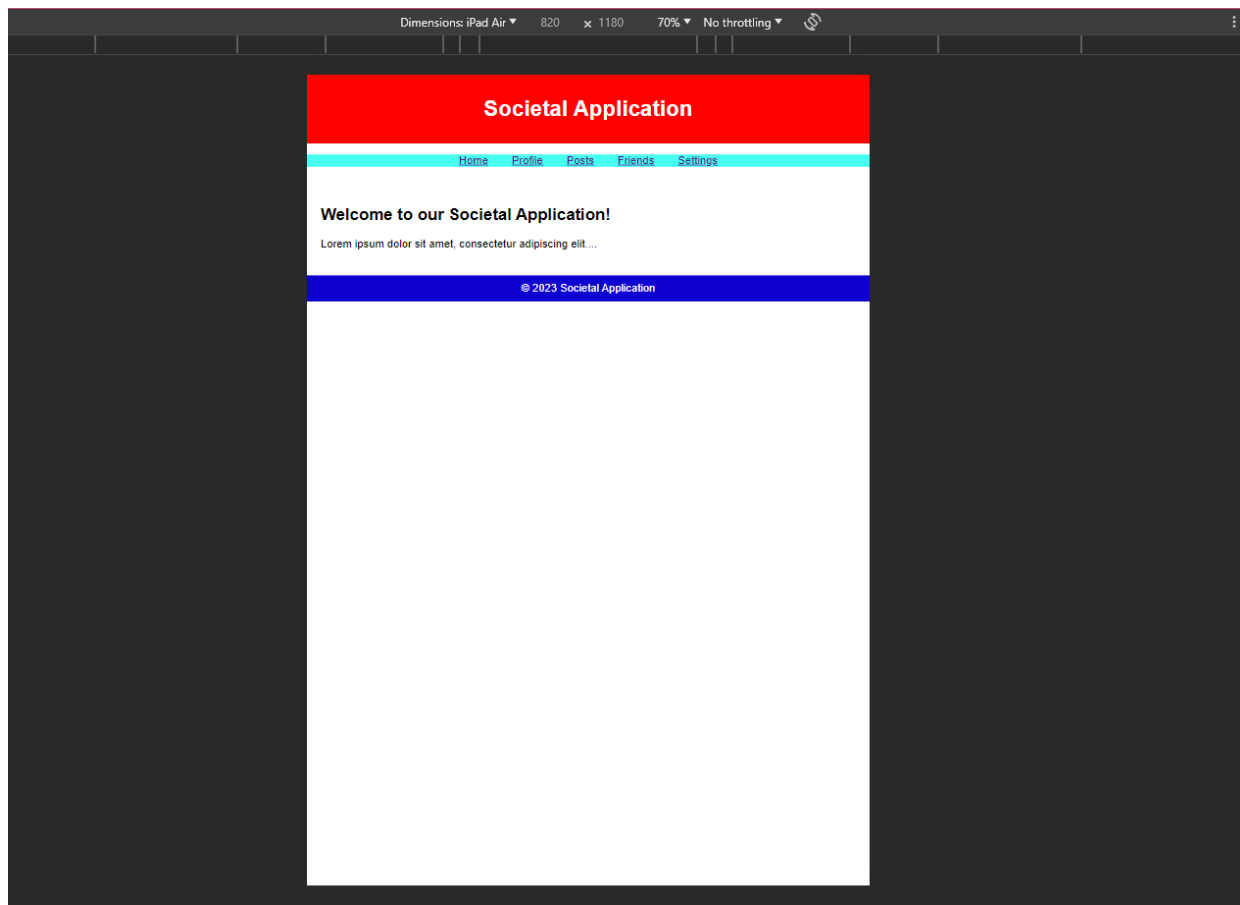
```

OUTPUT :

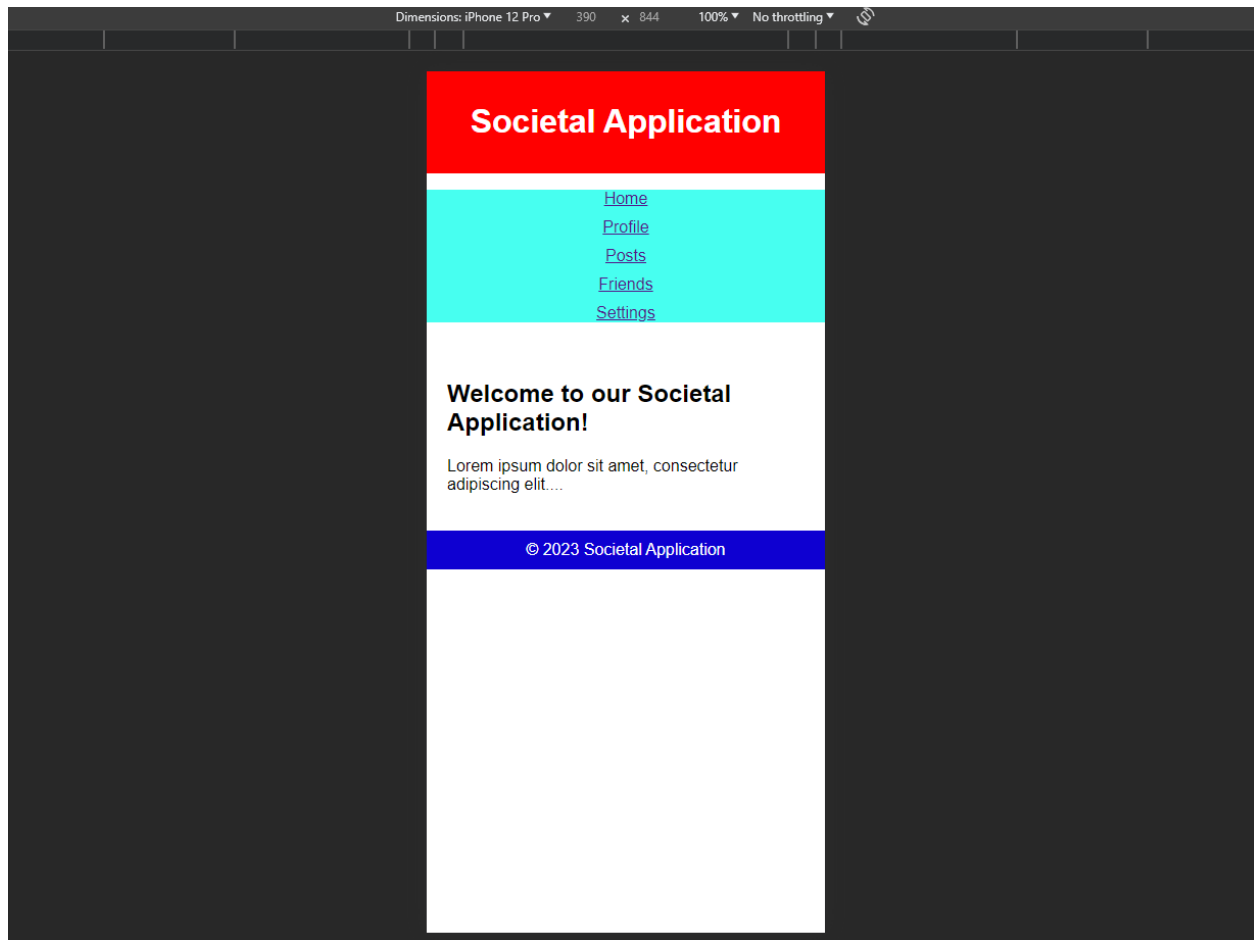
DESKTOP VIEW:



TABLET VIEW:



PHONE VIEW:



EX.02: Exploring various UI Interaction Patterns

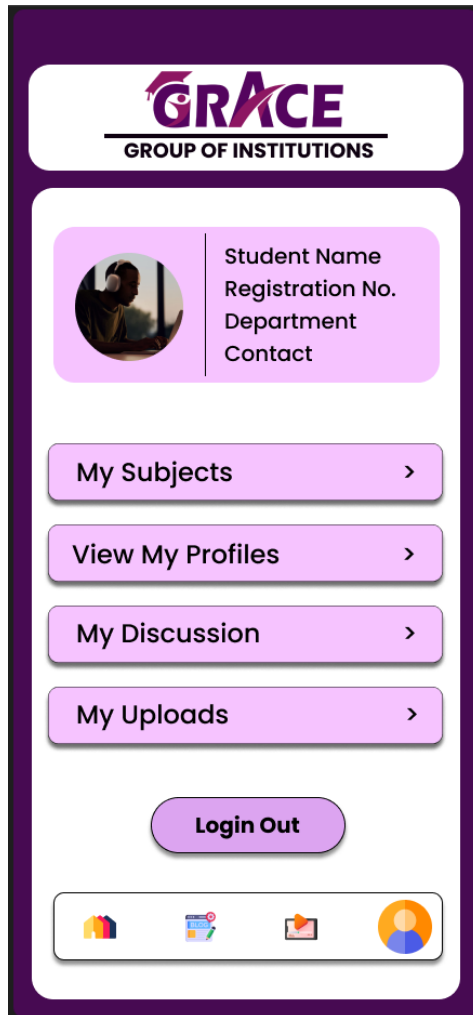
Aim: To explore various UI interaction patterns.

Algorithm/Procedure:

- Set objectives and understand user needs.
- Research and gather design inspiration.
- Create wireframes for layout and structure.
- Utilize Figma components and styles.
- Prototype interactions using Figma's features.
- Test your design with users for feedback.
- Iterate and refine based on feedback.
- Document your design decisions.

Figma Design:





Result: Thus various UI interaction patterns have been explored successfully.

Ex.03: Developing wireflow diagram for application using open-source software

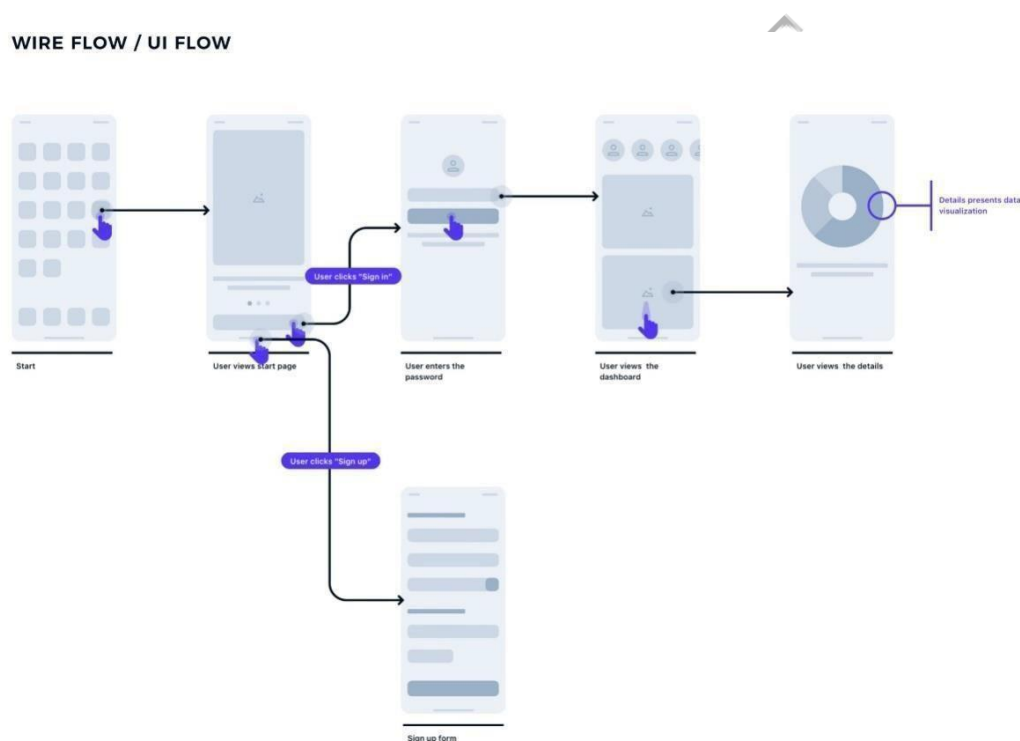
Aim: To develop Wireflow diagram for application using open-source software

Algorithm/Procedure:

1. **Define Purpose and Goals:** Determine the diagram's purpose and goals, focusing on user flows, navigation, and interactions.
2. **Identify User Personas:** If applicable, specify user personas for a user-centric approach.
3. **Gather Requirements:** Collect project information, including existing designs and functionality requirements.
4. **Select Software:** Choose open-source design software, such as Figma, for wireflow creation.
5. **Create a Project:** Begin a new project in your chosen software and set up the canvas to match your project's needs.
6. **Wireframe Screens:** Develop wireframes for each application screen, focusing on visual structure.
7. **Define Interactions:** Add interaction notes or links to illustrate navigation and user interactions.
8. **Create User Flows:** Connect wireframes to illustrate user journeys, navigation paths, and interactions.
9. **Add Annotations:** Include descriptions to clarify elements and interactions in each wireframe.
10. **Collaborate and Share:** Utilize collaboration features to gather feedback from team members and stakeholders.
11. **Iterate and Refine:** Revise the wireflow diagram based on - feedback, ensuring alignment with project goals.

12. **Finalize and Export:** Clean up the wireflow diagram and export it to a suitable format for sharing and documentation.
13. **Document the Wireflow:** Create a reference guide to explain the wireflow's purpose and key notes for stakeholders and developers.
14. **Maintain Consistency:** Keep the wireflow diagram in sync with the application's actual design, updating it as needed.

Design:



Result: Thus Wireflow diagram for application using open-source software has been developed successfully.

Ex.04: Hands on Design Thinking Process for a new product

Aim: To apply the design thinking process for a new product.

Algorithm/Procedure:

Empathize: Begin by conducting user research and interviews to gain insights into potential user needs and pain points related to smartphone usage.

Define: Analyze the gathered information to define a clear and specific problem statement. For example, "Users need a more efficient way to track their daily fitness activities."

Ideate: Organize brainstorming sessions with a diverse team to generate a wide range of creative solutions. Encourage free thinking and open collaboration.

Prototype: Create a low-fidelity prototype of the smartphone app. This can be a paper sketch or a digital wireframe that represents the app's basic functionality.

Test: Conduct user testing sessions with a small group of potential users. Observe how they interact with the prototype and gather feedback.

Iterate: Based on user feedback, refine the prototype and make necessary improvements to address user concerns or suggestions.

Prototype (Again): Create a more advanced prototype, closer to the final product. It should incorporate the changes and improvements identified during the initial testing phase.

Test (Again): Conduct another round of user testing, this time with a larger group of users. Gather data on usability, functionality, and overall user experience.

Refine: Analyze the results of the second testing phase and make further refinements to the app design and functionality.

Implement: Develop the final version of the smartphone app, incorporating all the changes and improvements identified during the design thinking process.

Test (Final Testing): Conduct thorough testing of the fully developed app to ensure it's bug-free and ready for launch.

Launch: Release the app to the target market, accompanied by marketing and promotion efforts.

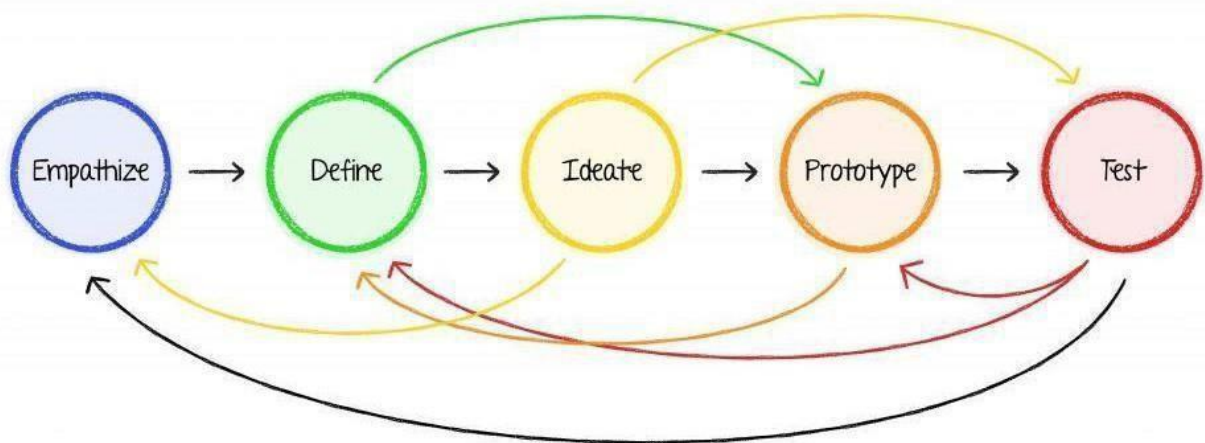
Example:

Let's say the team is designing a fitness tracking app. During the "Empathize" phase, they conduct interviews and surveys with potential users, discovering that users find it challenging to keep track of their daily physical activities. In the "Define" phase, they define the problem as "Users need a more efficient way to track their daily fitness activities."

In the "Ideate" phase, the team generates multiple ideas, including features like GPS tracking, step counting, and customizable fitness goals. They create a low-fidelity prototype that represents these features. In the first round of user testing, they observe that users have difficulty navigating the app.

After gathering feedback and identifying navigation issues, the team iterates by redesigning the user interface to improve user experience. They create an advanced prototype with a more intuitive interface and test it with a larger group of users in the second round of testing.

Based on this testing, the team further refines the app, making sure it's user-friendly, bug-free, and meets the needs of the target audience. Finally, they implement and launch the fitness tracking app to help users easily track their daily activities.



Design Thinking

Result: Thus the design thinking process for new product has been studied.

Ex.05: Brainstorming feature for proposed product

Aim:

The aim of this process is to generate innovative and practical feature ideas for a proposed product, ensuring that the final product meets user needs, addresses pain points, and has a competitive edge in the market.

Algorithm/Procedure:

Understand the Product Concept:

Begin by thoroughly understanding the proposed product's concept, its target audience, and its unique selling points.

Gather a Diverse Team:

Assemble a cross-functional team with members from various departments (e.g., product development, marketing, design) to bring different perspectives to the brainstorming session.

Set Clear Objectives:

Define clear objectives for the brainstorming session. What problems should the new features solve? What goals should they achieve?

Warm-Up and Icebreaker:

Start the session with a warm-up or icebreaker activity to encourage creative thinking and open communication within the team.

Idea Generation:

Allow team members to freely brainstorm feature ideas. Encourage a "no idea is a bad idea" mindset. Use techniques like mind mapping, brainstorming software, or post-it notes on a whiteboard to record ideas.

Categorize and Prioritize:

Group similar ideas together, and prioritize them based on factors like feasibility, potential impact, and alignment with the product concept.

SWOT Analysis:

Conduct a SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis for each feature idea to evaluate its potential in the market.

Feasibility Assessment:

Assess the technical, financial, and resource feasibility of implementing the proposed features.

Market Research:

Conduct market research to identify user preferences and gather insights that can inform feature development.

Prototype and User Testing:

Create prototypes or mockups of the proposed features and conduct user testing to gather feedback and refine the ideas.

Cost-Benefit Analysis:

Evaluate the expected cost of development against the projected benefits, such as increased user engagement, retention, or revenue.

Risk Assessment:

Identify potential risks associated with each feature and develop mitigation strategies.

Finalize Feature Set:

Based on the assessment, finalize the set of features to be included in the product. Ensure they align with the product's vision and goals.

Documentation:

Document the chosen features, their objectives, and the rationale behind their selection. This document will guide the development team.

Iterate as Needed:

Keep an open line of communication for ongoing feature refinements and iterations, especially as more data and insights become available.

Example:

Suppose a software company is developing a new mobile messaging app. During the brainstorming session, the team generates a wide range of feature ideas, including:

End-to-End Encryption: To ensure user privacy and data security.

Message Scheduling: Allowing users to schedule messages to be sent at a specific time.

Reaction Emojis: A feature that lets users react to messages with emojis for more expressive communication.

Dark Mode: A night-friendly theme for the app.

Polls and Surveys: Integration of polls and surveys within the chat for easy decision-making.

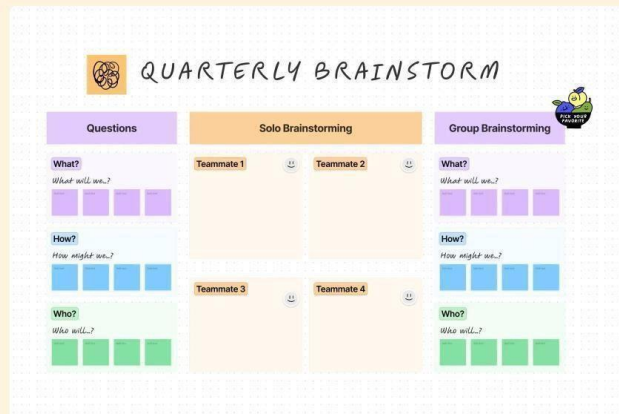
Auto-Translate: Real-time language translation for international communication.

Get started with an online brainstorming template

Save time and lead more engaging brainstorms with hundreds of ready-made templates.

* Quick brainstorm

- Crazy 8s
- Four actions framework
- Reverse brainstorming
- Retrospective
- Roadmap planning

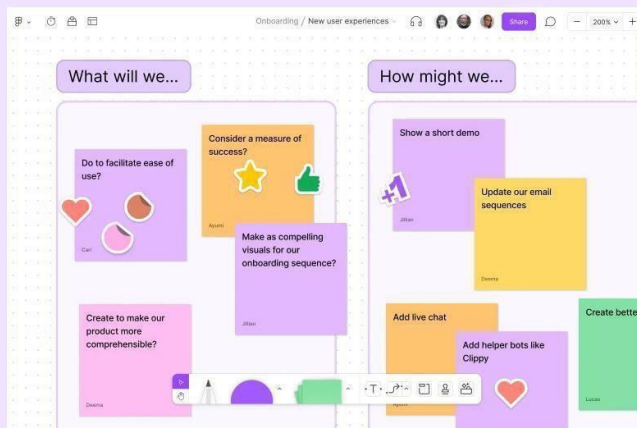


Start with this template

See all brainstorm templates

Build on ideas together in real time

Run online brainstorms that empower everyone to contribute so that you can generate better creative solutions—together.



Give every team a say

Create a space where teams can ideate and create together.

Result: Thus brainstorming feature for proposed product has been applied and executed successfully.

Ex.6: Identify a customer problem to solve

Aim: The aim of this experiment is to identify a customer problem to solve effectively, which is crucial for product development, customer satisfaction, and business success.

Algorithm/Procedure:

Customer Segmentation:

Begin by segmenting your customer base into different groups based on demographics, behavior, or other relevant criteria.

Data Collection:

Gather data from these customer segments through surveys, interviews, feedback forms, and analytics tools. You can also utilize data from your customer support system, website, or app analytics.

Problem Identification Metrics:

Define key metrics and indicators to identify customer problems. Examples include high bounce rates on a specific webpage, low customer satisfaction scores, or a surge in support tickets related to a specific issue.

Data Analysis:

Analyze the collected data to identify patterns, trends, and common issues reported by customers. Data analysis tools and techniques, such as data mining or sentiment analysis, can be useful.

Prioritization:

Prioritize the identified problems based on their impact on customers and your business. You can use techniques like the Moscow method (Must-haves, Should-haves, Could-haves, Won't-haves) to prioritize.

Root Cause Analysis:

Conduct a root cause analysis for each identified problem. Understand why these issues are occurring by delving into the underlying causes.

Solution Ideation:

Brainstorm potential solutions for the identified problems. Encourage cross-functional teams to contribute ideas and consider how these solutions align with your business goals.

Experiment Design:

Design controlled experiments or A/B tests to validate the proposed solutions. Ensure you have a clear hypothesis and success criteria for each experiment.

Implementation:

Implement the proposed solutions on a small scale to observe their impact. This might involve website changes, process adjustments, or feature additions.

Data Collection Post-Implementation:

Continue to collect data after implementing the solutions to assess their effectiveness. Monitor key metrics to see if they improve.

Analysis and Validation:

Analyze the post-implementation data to validate whether the proposed solutions have effectively addressed the customer problem. Make data-driven decisions.

Feedback and Iteration:

Collect feedback from customers regarding the changes and iterate on the solutions based on their input. Continuous improvement is key.

Example:

Let's say you're running an e-commerce platform and want to identify a customer problem related to checkout abandonment. Here's how you could apply the algorithm/procedure:

Customer Segmentation:

Segment customers based on their demographics and purchase history.

Data Collection:

Gather data through customer surveys, web analytics, and feedback forms.

Problem Identification Metrics:

One of your key metrics is the high rate of customers abandoning their shopping carts before completing the purchase.

Data Analysis:

Analyze the data and find that a significant number of customers abandon their carts at the payment stage.

Prioritization:

Prioritize the payment abandonment issue because it directly affects revenue.

Root Cause Analysis:

Discover that complex payment options and a lack of guest checkout are causing the problem.

Solution Ideation:

Brainstorm solutions, including simplifying payment options and adding a guest checkout feature.

Experiment Design:

Design A/B tests to measure the impact of these changes on cart abandonment rates.

Implementation:

Implement the proposed solutions on a small scale for testing.

Data Collection Post-Implementation:

Collect data on cart abandonment rates after the changes are implemented.

Analysis and Validation:

Analyze the post-implementation data and find that cart abandonment rates have significantly decreased.

Feedback and Iteration:

Collect feedback from customers who completed purchases and continue to iterate on the checkout process to further enhance the customer experience and address any remaining issues.

Result: Thus a customer problem was identified and understood successfully.

Ex.7: Conduct end-to-end user research - User research, creating personas, Ideation process (User stories, Scenarios), Flow diagrams, Flow Mapping.

Aim:

The aim of this experiment is to conduct end-to-end user research and design process to develop a user-centered solution for a specific problem. This process includes user research, creating personas, ideation (user stories, scenarios), and creating flow diagrams and flow maps.

Algorithm/Procedure:

Define the Problem:

Clearly define the problem or challenge you want to address through this user-centered design process.

User Research:

Conduct user interviews, surveys, or observations to gather insights and data about the target users. Analyze the collected data to identify user needs, pain points, and behaviors.

Create Personas:

Based on the research findings, create user personas. Personas are fictional representations of your typical users, including their goals, needs, and characteristics.

Ideation:

Brainstorm creative ideas to address the identified problems and fulfill user needs. Develop user stories and scenarios to articulate how users will interact with the proposed solution.

User Stories:

Create user stories using the "As a [type of user], I want [an action] so that [benefit/value]" format. User stories should capture specific user tasks and their motivations.

Scenarios:

Develop detailed narratives (scenarios) that illustrate how users will use the solution to achieve their goals. Scenarios should provide context, user actions, and expected outcomes.

Flow Diagrams:

Create flow diagrams to visualize the user's journey through the solution.

Use symbols and arrows to represent user actions, system responses, and transitions between different screens or stages.

Flow Mapping:

Develop flow maps to provide a holistic view of the user experience. Connect user stories and scenarios to specific steps in the flow, highlighting decision points and potential pain points.

Iterate and Test:

Review and refine your user stories, scenarios, flow diagrams, and flow maps based on feedback from stakeholders or potential users. Conduct usability testing to validate the proposed user experience.

Implementation:

Work with developers and designers to implement the user-centered solution based on the finalized flow and design.

Evaluation:

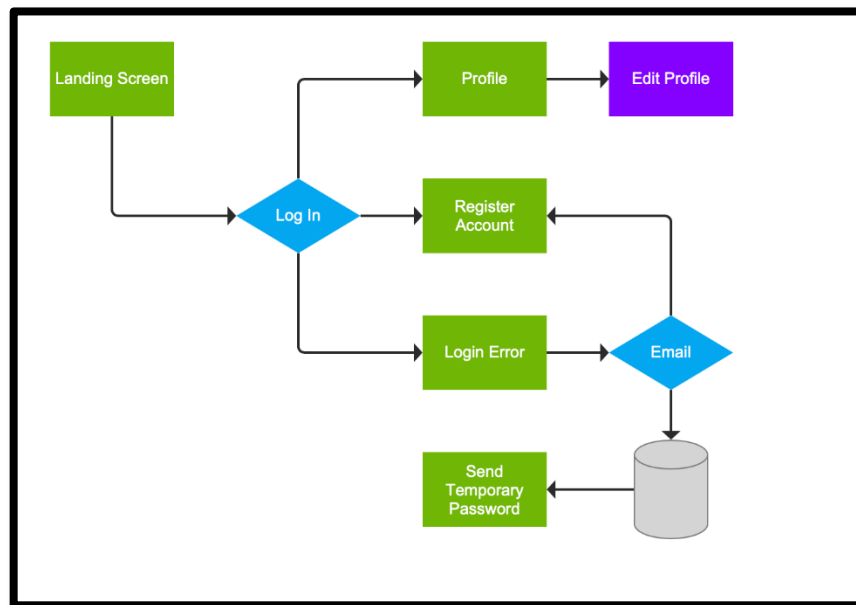
After the solution is implemented, evaluate its effectiveness by gathering user feedback and monitoring key performance metrics.

Iterate and Improve:

Continuously iterate on the design and user experience based on user feedback and changing needs.

Design:

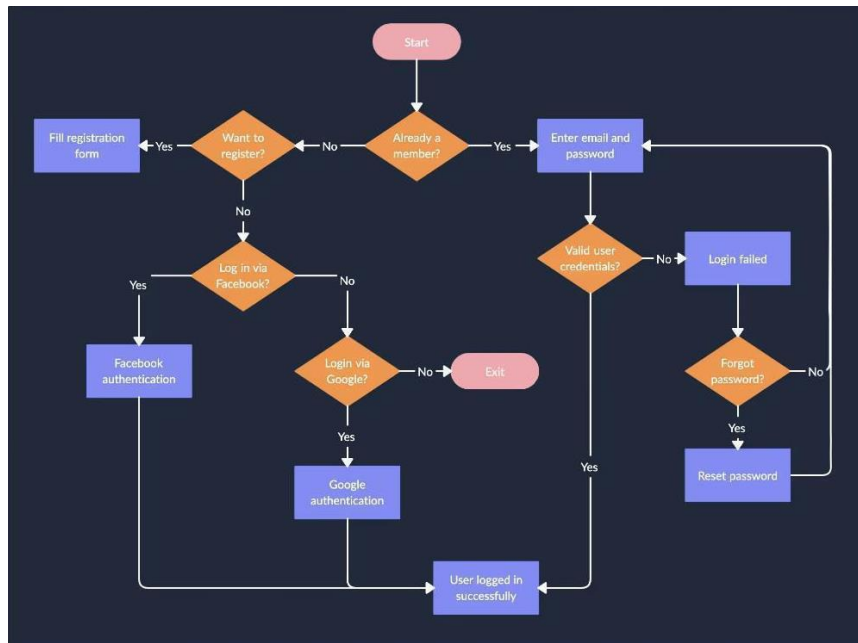
Flow Diagram:



User Personas:

The form is titled 'CAREERFOUNDRY' and includes a 'photo' placeholder. It contains several input fields for user information: 'Name:', 'Age:', 'Job:', 'Location:', 'Family status:', and 'My Story:'. Below these are four sections for user needs and goals, each with a list of bullet points: 'Needs:', 'Frustrations:', 'Goals:', and 'Enviroment:'. The form is designed to collect detailed user information for creating a user persona.

Flow Mapping:



Result:

Thus a user-centered problem was identified to solve through a comprehensive process of user research, persona creation, ideation (including user stories and scenarios), flow diagrams, and flow mapping, with the goal of successfully addressing user needs and delivering an exceptional user experience.

Ex.8: Sketch, design with popular tool and build a prototype and perform usability testing and identify improvements

Aim:

The aim of this experiment is to design a user-friendly mobile app for task management, create a prototype using a popular design tool, perform usability testing, and identify improvements to enhance the user experience.

Algorithm/Procedure:

Define Objectives and User Persona:

Define the objectives of the task management app. Create a user persona to represent the target audience.

Sketch and Wireframe:

Start with sketching the basic layout and functionality of the app on paper or digitally. Create low-fidelity wireframes to visualize the app's structure and layout.

Design with a Popular Tool:

Choose a popular design tool such as Adobe XD, Sketch, Figma, or InVision. Create high-fidelity designs with attention to visual elements, typography, and color schemes. Implement the user interface (UI) based on best practices and your user persona's preferences.

Prototype Creation:

Use the design tool to create interactive prototypes with clickable elements and transitions. Ensure that the prototype represents the app's core functionalities.

Recruit Participants for Usability Testing:

Identify potential users or participants who match the user persona. Prepare a usability testing plan, including tasks to be performed within the prototype.

Usability Testing:

Conduct usability testing sessions with participants. The participants are asked to perform specific tasks within the prototype. Observe and record their interactions and gather feedback on their experience.

Analyze and Identify Improvements:

Analyze the usability testing data to identify pain points and areas of improvement. Look for common patterns and issues encountered by users.

Iterate on the Design:

Implement the necessary design improvements based on the feedback received. Make changes to the prototype to address identified issues.

Second Round of Usability Testing:

Conduct a second round of usability testing with new or the same participants to evaluate the impact of the design improvements.

Finalize the Prototype:

Make any final adjustments based on the results of the second usability testing round.

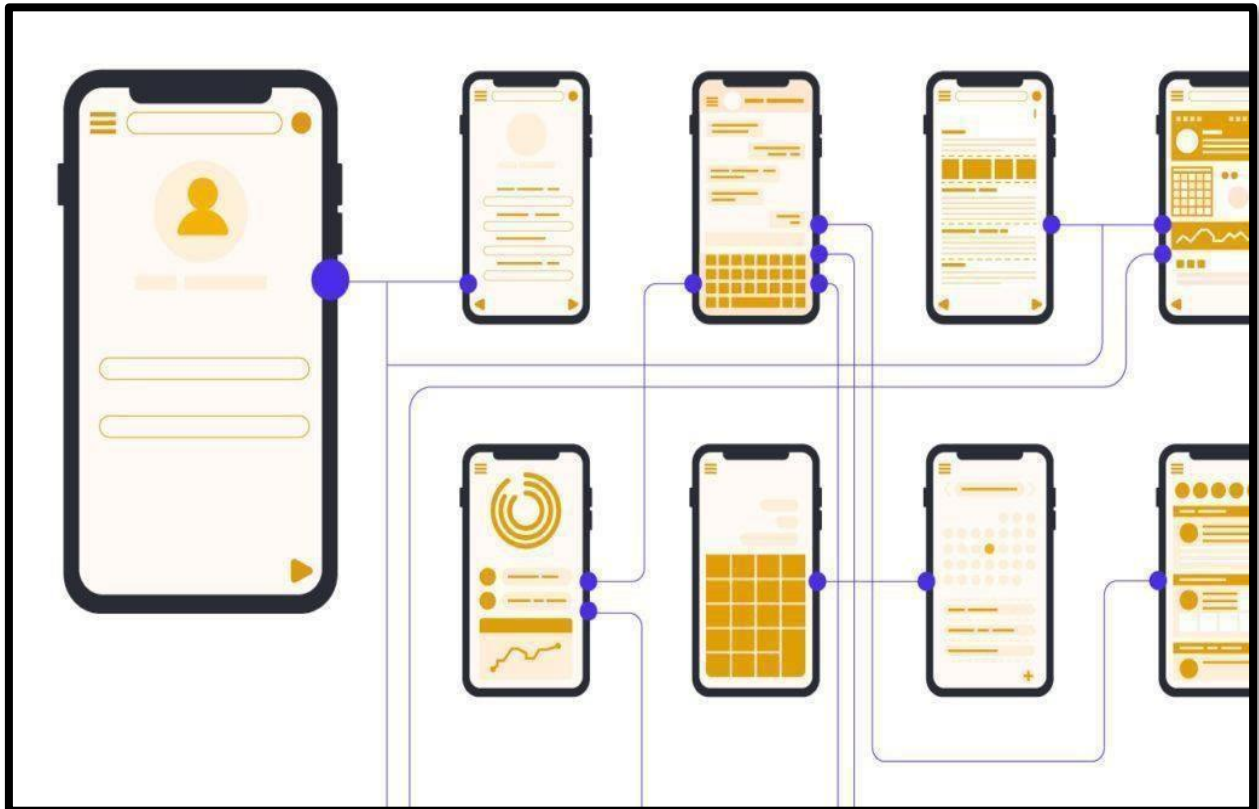
Document Findings and Recommendations:

Document the findings from both rounds of usability testing.

Provide clear recommendations for further improvements or development.

Conclusion:

Conclude the experiment by summarizing the improvements made to the prototype and how they enhance the user experience.

Design:**Result:**

Thus, Sketching, building a prototype, performing usability testing and identifying improvements has been executed successfully.