



**CHENNAI
INSTITUTE OF TECHNOLOGY**
(Autonomous)



NIRF
151 - 200 Band
Engineering 2023



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BE - Computer Science and Engineering

(AIML, Cyber Security, CSBS)

Anna University Regulation: 2021

UI AND UX DESIGN

III Year / V Semester

LAB MANUAL

Prepared By,

Mrs.Sumathi S

Asst. Professor

EX.No:1

Date:

DESIGNING A RESPONSIVE LAYOUT FOR A SOCIETAL APPLICATION

Figma:

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

Algorithm/Procedure:

Creating a responsive website in Figma:

1. Define your website's purpose and audience.
2. Set breakpoints for different screens.
3. Design desktop version.
4. Use grids for layouts.
5. Design for tablet and mobile.
6. Create responsive components.
7. Apply constraints to elements.
8. Use relative font sizes.
9. Optimize images.
10. Test and prototype.
11. Use grid/flexbox layouts.
12. Design responsive navigation.
13. Consider touch interactions.
14. Test on various devices.
15. Iterate based on feedback.
16. Document choices.
17. Share with developers.
18. Maintain and update design.

UI Design:

Result: Thus, designing of responsive layout for a societal application has been performed successfully.

EX. No:2

Date:

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. No:3

Date:

DEVELOPING AN INTERFACE WITH PROPER UI STYLE GUIDES

Aim: To develop an Interface with Proper UI Style Guides.

Algorithm/Procedure:

- Define Goals and Audience: Understand the project's purpose and target users.
- Research and Inspiration: Gather industry insights and design inspiration.
- Create a Figma Project: Start a new Figma project.
- Workspace Setup: Organize Figma files and create sections for style guides and components.
- Brand Guidelines: Set color, typography, and brand-related guidelines.
- UI Components: Create component libraries for buttons, forms, icons, and navigation.
- Typography: Define font styles, sizes, and spacing.
- Iconography: Design and organize icons as components.
- Color System: Document primary, secondary, background, and text colors.
- Grids and Layouts: Establish grid systems for different devices.
- Accessibility Guidelines: Ensure WCAG-compliant contrast and text sizing.
- UI Elements: Document usage guidelines for various UI elements.

- Wireframes: Create wireframes and user flows for screen layout and interaction.
- Visual Design: Develop screens based on wireframes using components and styles.
- Interactive Prototyping: Build interactive prototypes for user testing and feedback.

Figma Design:

Result: Thus, an Interface with Proper UI Style Guides has been developed successfully.

Ex. No:4

Date:

DEVELOPING WIREFLOW DIAGRAM FOR APPLICATION USING OPEN-SOURCE SOFTWARE

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

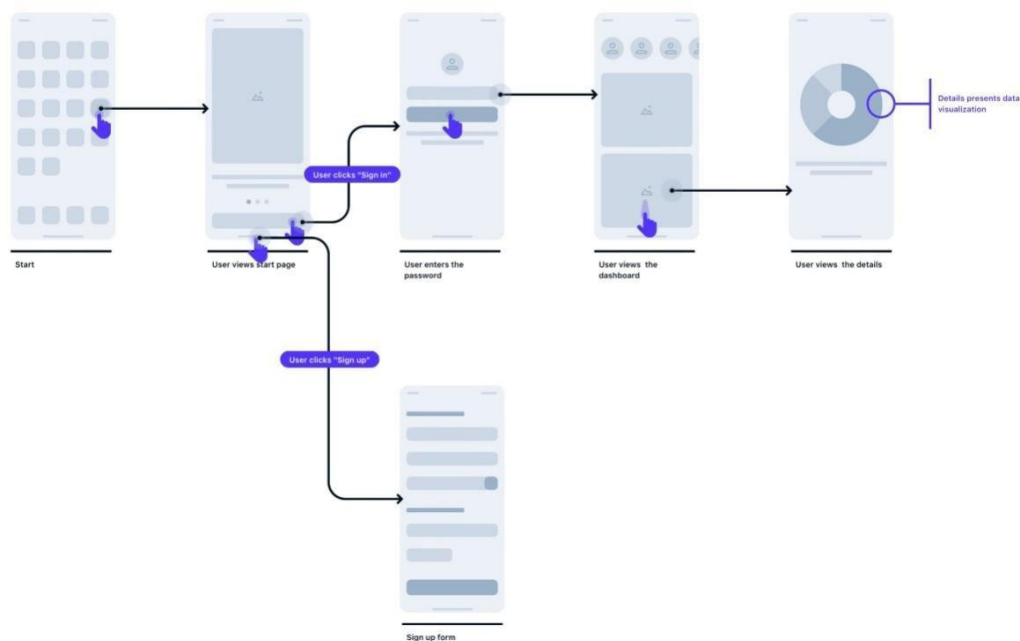
Algorithm/Procedure:

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:

WIRE FLOW / UI FLOW



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

Ex. No:5

Date:

EXPLORING VARIOUS OPEN-SOURCE COLLABORATIVE INTERFACE PLATFORM

Aim: To Explore Various Open-Source Collaborative Interface Platform.

Algorithm/Procedure:

1. **Needs Assessment:** Define your team's requirements and goals.
2. **Research Platforms:** Identify open-source collaborative tools.
3. **Feature Comparison:** Assess features and compatibility.
4. **Community Support:** Check for active communities.
5. **Installation:** Set up the chosen platform.
6. **User Training:** Train and encourage team adoption.
7. **Security:** Ensure data security and privacy.
8. **Integration:** Check for compatibility with existing tools.
9. **Testing and Feedback:** Pilot testing and gather feedback.
10. **Scalability:** Ensure the platform can grow with your team.
11. **Documentation:** Create user resources and guides.
12. **Community Engagement:** Utilize community support.
13. **Maintenance:** Keep the platform updated and secure.
14. **Feedback Loop:** Encourage ongoing user feedback.
15. **Legal Compliance:** Ensure adherence to licensing and Legal Requirements.
16. **Backup and Recovery:** Implement data safety measures.
17. **Periodic Evaluation:** Continuously assess platform suitability.
18. **Migration Plan:** Prepare for possible platform changes.
19. **Success Sharing:** Promote successful platform usage.

Design:

Result: Thus, various open-source collaborative interfaces Platform has been explored successfully.

Ex. No:6

Date:

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.

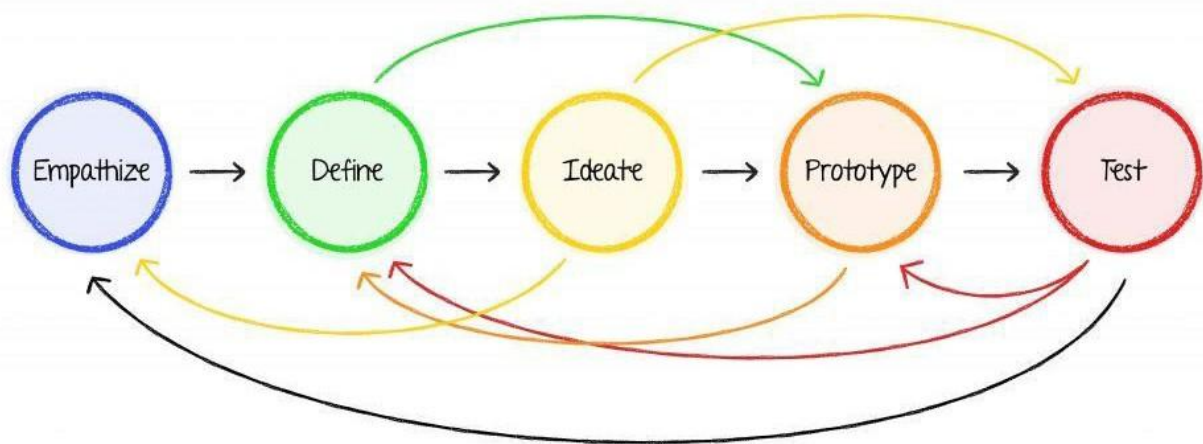
Example:

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. No:7

Date:

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

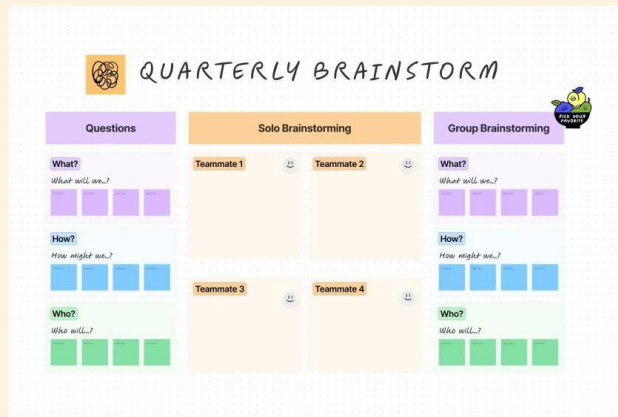
Crazy 8s

Four actions framework

Reverse brainstorming

Retrospective

Roadmap planning

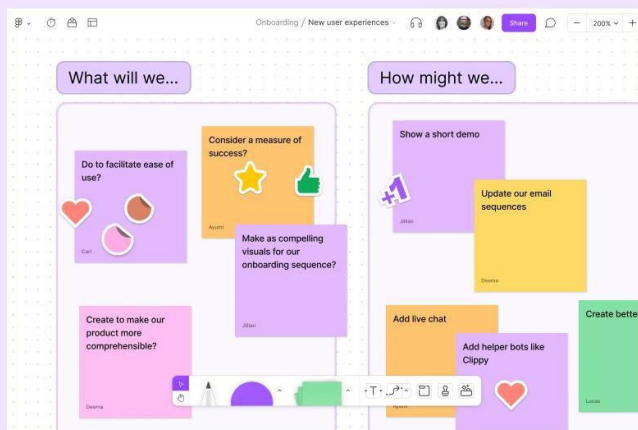


Start with this template

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. No:8

Date:

DEFINING THE LOOK AND FEEL OF THE NEW PROJECT

Aim:

The aim is to establish the visual design and user experience for a new project, ensuring it aligns with the project's goals and provides an appealing, intuitive, and cohesive interface for users.

Algorithm/Procedure:

Project Goal Assessment:

Understand the project's objectives, target audience, and scope. This sets the foundation for design decisions.

Research and Inspiration:

Gather inspiration from existing designs and industry trends. Create mood boards or design boards to collect visual references.

Define Design Principles:

Determine the core design principles that will guide the project's look and feel. These could include simplicity, consistency, accessibility, and branding.

Wireframing and Prototyping:

Create wireframes or low-fidelity prototypes to plan the layout and structure of the user interface. Use tools like Figma, Sketch, or Adobe XD for digital projects.

Visual Design:

Develop a color palette, typography choices, and graphic elements (icons, images, logos) that reflect the project's identity. Create high-fidelity designs using design software.

User Interaction Design:

Define user interactions and behaviors, including animations, transitions, and micro-interactions. Ensure a smooth and intuitive user experience.

Responsive Design:

Adapt the design to various screen sizes and devices, focusing on mobile responsiveness.

Accessibility and Usability Testing:

Evaluate the design for accessibility, ensuring it's usable by individuals with disabilities. Conduct usability testing with potential users to gather feedback.

Iteration and Feedback:

Refine the design based on feedback from users and stakeholders.

Documentation:

Create design documentation that includes guidelines for developers to implement the design.

Development Integration:

Collaborate with developers to ensure the design is implemented accurately in the project.

User Testing:

Conduct user testing with real users to verify the design's effectiveness.

Finalization:

Make necessary adjustments and finalize the design elements for deployment.

Example:

Imagine you're tasked with defining the look and feel of a new e-commerce website. You go through the following steps:

Assess the project's goal: To create a user-friendly, visually appealing online store. Research e-commerce design trends and competitor websites.

Define design principles:

Simplicity, trustworthiness, and a focus on the product. Create wireframes for key pages like the homepage, product listing, and product detail pages. Develop a color palette featuring subtle, trust-inducing colors.

Choose a clear and readable typography for product descriptions and headings. Plan user interactions like product image zoom and streamline checkout process.

Ensure that the design is responsive and mobile-friendly. Conduct accessibility testing to meet WCAG standards.

Iterate on the design based on usability testing feedback.

Document the design guidelines for the development team.

Collaborate closely with developers to bring the design to life.

Conduct user testing to validate the design's effectiveness. Make final adjustments and prepare for the website's launch.

Design:

Result: Thus, the Look and Feel of the new Project has been defined successfully.

Ex. No:9

Date:

**CREATE A SAMPLE PATTERN LIBRARY FOR THE
PRODUCT (MOOD BOARD, FONTS, COLORS
BASED ON UI PRINCIPLES)**

Aim: The aim of this project is to create a sample Pattern Library for a product that includes mood boards, fonts, and color schemes based on UI (User Interface) principles. The Pattern Library will serve as a design reference for maintaining consistency and cohesion in the product's user interface.

Algorithm/Procedure:

Define the Scope:

Identify the product for which you are creating the Pattern Library. Understand the product's target audience, brand identity, and design goals.

Gather Inspiration:

Research existing UI designs, competitor products, and design trends to gather inspiration. This will help you create a mood board that captures the desired aesthetics.

Create Mood Boards:

Using design software or tools like Adobe XD or Figma, create mood boards that reflect the desired visual style, mood, and tone of the product. Include images, colors, typography samples, and other visual elements that represent the brand and design principles.

Choose Fonts:

Select fonts that align with the product's brand and design goals. Consider readability, hierarchy, and scalability. Choose a primary font for headings and body text, and possibly secondary fonts for accents.

Define Color Palette:

Create a color palette that includes primary, secondary, and accent colors. Ensure the colors are harmonious and comply with accessibility guidelines. Provide color codes (hex, RGB, or HSL) for each color.

Document UI Principles:

Document the UI design principles that underpin the Pattern Library. This may include guidelines on spacing, layout, typography, and interaction behaviors.

Organize and Label:

organize the Pattern Library in a clear and accessible manner. Label each component, color, and typography choice, and provide context or usage guidelines.

Mockup Example Screens:

Create example screens or wireframes using the fonts, colors, and components defined in the Pattern Library. These screens should showcase the design principles in action.

Test and Refine:

Share the Pattern Library with designers, stakeholders, or users for feedback. Make refinements based on the feedback to ensure it aligns with the project's goals.

Maintenance and Version Control:

As the product evolves, maintain and update the Pattern Library to reflect any design changes. Use version control systems to keep track of changes and updates.

Example:

Let's say you're creating a Pattern Library for a mobile app focused on wellness and meditation. The aim is to create a soothing and user-friendly design that promotes relaxation. Here's a simplified example:

Mood Board: Create a mood board with images of serene landscapes, calming color schemes (e.g., blues and greens), and sample UI elements that convey a sense of tranquility.

Fonts: Choose "Poppins" as the primary font for headings and "Roboto" for body text. These fonts are clean, readable, and complement the app's aesthetic.

Color Palette: Define a color palette that includes calming colors like "#3CBDBE" for primary elements, "#70C1B3" for secondary elements, and "#EFEFF0" for background. Ensure that these colors meet accessibility standards.



graceful **DARING** informal
wistful contemporary
hand-crafted **authoritative**
friendly **PLAYFUL** personal
trustworthy neutral **STRONG**

Moodboards

mm

LOVE IT.

Design:

Result: Thus, a sample pattern library for a product was created successfully.

Ex. No:10

Date:

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, customersatisfaction, 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.

Design:

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

Ex. No:11

Date:

**CONDUCT END-TO-END USER RESEARCH - USER RESEARCH,
CREATING PERSONAS, IDEATION PROCESS (USER STORIES,
SCENARIOS), FLOW DIAGRAMS, FLOWMAPPING**

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:

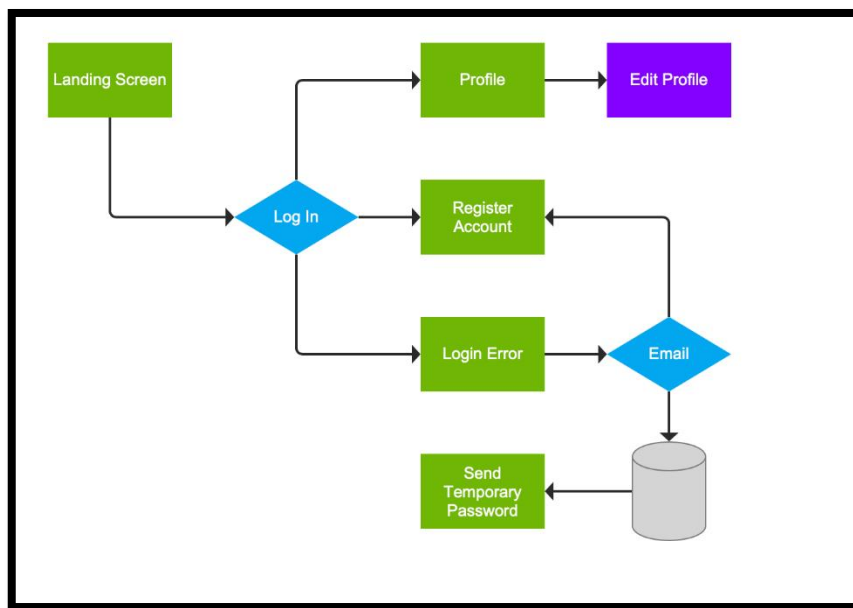
After the solution is implemented, evaluate its effectiveness by gathering user feedback and monitoring key performance metrics.

Iterate and Improve:

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

Design:

Flow Diagram:



User Personas:

CAREERFOUNDRY

photo

Name: Location:

Age: Family status:

Job:

My Story:

Quote:

Needs:

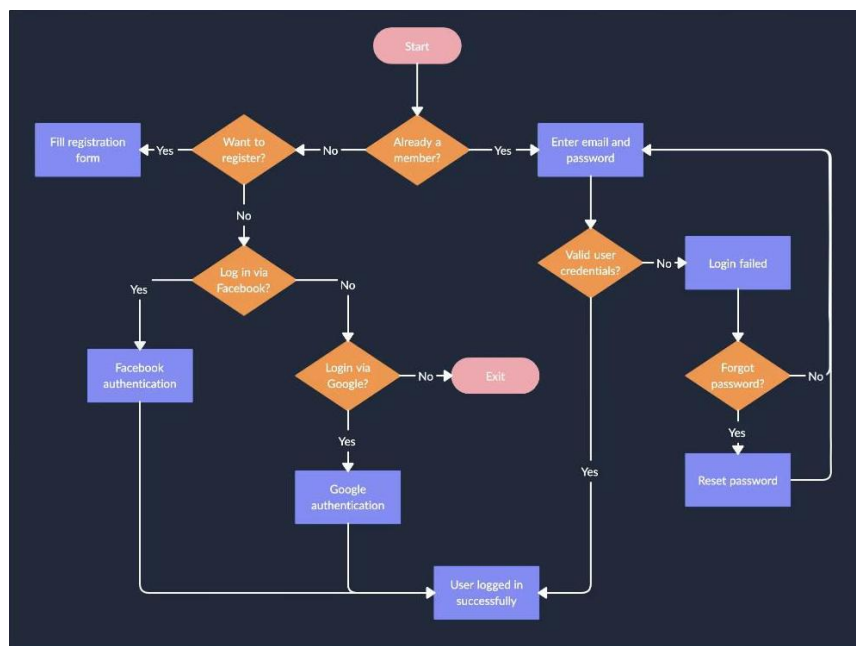
Frustrations:

Goals:

Environment:

circle primary devices

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.

ExNo:12

Date:

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 recommendation 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.