# Goal Report
# On
# Cashify Web and Console Feature

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

| | |
|---|---|
| **Faculty Mentor:** | **Submitted by:** |
| Mr. Saif Nalband | Vartika Gautam |
| | (102103397) |
| **Industrial Mentor:** | |
| Mr.Hemchand Yadav | |

**Computer Science and Engineering Department**
**Thapar Institute of Engineering & Technology, Patiala**

## Problem faced:

At Cashify, multiple pods work on different features and flows, leading to inconsistent UI implementations across the platform. Without a unified system to preview and validate UI changes, teams often face design inconsistencies, duplication of efforts, and late discovery of visual bugs. This impacts user experience, slows down development, and increases maintenance overhead.

## Tentative Problem(s)/Objective(s):

1. **Lack of a Unified Design System Preview**
   Developers across pods do not have a centralized platform to view how common UI elements—such as buttons, fonts, and radio buttons—should appear and behave. This leads to confusion and inconsistency.

2. **Inconsistent Implementation of Components**
   In the absence of a clear visual reference, similar components are often implemented differently by various pods, resulting in a fragmented and inconsistent user interface across the application.

3. **No Pre-Integration Visual Validation**
   Teams push UI changes without having a mechanism to preview how global styles (e.g., fonts, button colors) may affect other modules. This leads to unintended side effects and broken layouts.

4. **Inefficient Testing and Collaboration**
   QA and design teams struggle to validate UI changes, as they need to manually inspect multiple flows. This increases testing time and the chances of missing visual issues.

5. **Delayed Bug Discovery**
   Without an early visual feedback system, UI bugs and regressions are often discovered late in the development or release cycle, leading to increased rework and delays.

# Proposed Solution(s):

### 1. View Standardized UI Components

- Showcase core UI elements such as buttons, inputs, radio buttons, typography, chips, etc.

- Allow preview of components in different states: **hover, active, and disabled**.

- Support **light and dark theme toggles** for better design testing.

### 2. Test UI Changes in Isolation

- Provide a sandbox environment for pods to test global style/token changes before pushing to production.

- Enable live editing of variables like --primary-color, --font-size-base, etc., with **real-time visualization** of effects across components.

### 3. Pod-Specific Modules

- Allow each pod to integrate and preview **custom UI components** specific to their flows (e.g., pricing cards, time slot selectors).

- Ensure modular structure while preserving a **unified design philosophy** across all teams.

### 4. Versioned Component Library

- Maintain a version history of all shared components to track updates.

- Provide easy access to linked design documentation (e.g., **Figma files**, component stories) for reference and collaboration.

### 5. Responsive Testing Support

- Include device toggle options to test components across **mobile, tablet, and desktop** breakpoints.

- Ensure that all UI elements are **responsive and consistent** across various screen sizes.