

# **BuildForGehu'25 Proposal**

## **SimpleUI**

**Version : v1.0.0**

**Team Tech Titans**

**Team Members:**

Suraj Sharma

Bhavesh Tripathi

Sourabh Singh

Kartikey Wariyal

Contact email: [Surajsharma60923@gmail.com](mailto:Surajsharma60923@gmail.com)

# Problem Statement

## The Problem:

The modern UI libraries like Bootstrap and Tailwind have transformed the frontend development but with some limitations.

### Limitations of Bootstrap:

- **Limited Customization** : The pre-designed components and grid system of Bootstrap sometimes leads to challenge of creating unique designs.
- **Similar Looks** : Using the default Bootstrap's components can lead to websites with similar look to others if no significant customization is made.

### Limitations of Tailwind :

- **No prebuilt components** : Tailwind doesn't provide any ready to use components ,leading to design everything from scratch.
- **Learning Curve** : The utility first approach and extensive class libraries of Tailwind is challenging for beginners.

## Addressed Gap:

These problem leads to a need of a hybrid UI library that :

- Offers customizable components out of the box.
- Prioritizes accessibility and performance from the root level.
- Must be suitable for beginners as well as developers and who want to build quick and clean designs.
- Supports interaction without requiring heavy tooling

# Proposed Solution

## The Solution :

SimpleUI is a lightweight, responsive CSS framework built from scratch to help developers build consistent, accessible, and beautiful UIs without relying on heavy libraries like Bootstrap or Tailwind. It is based on mobile first , component based architecture offering essential utility class for layout and styling .

## Features:

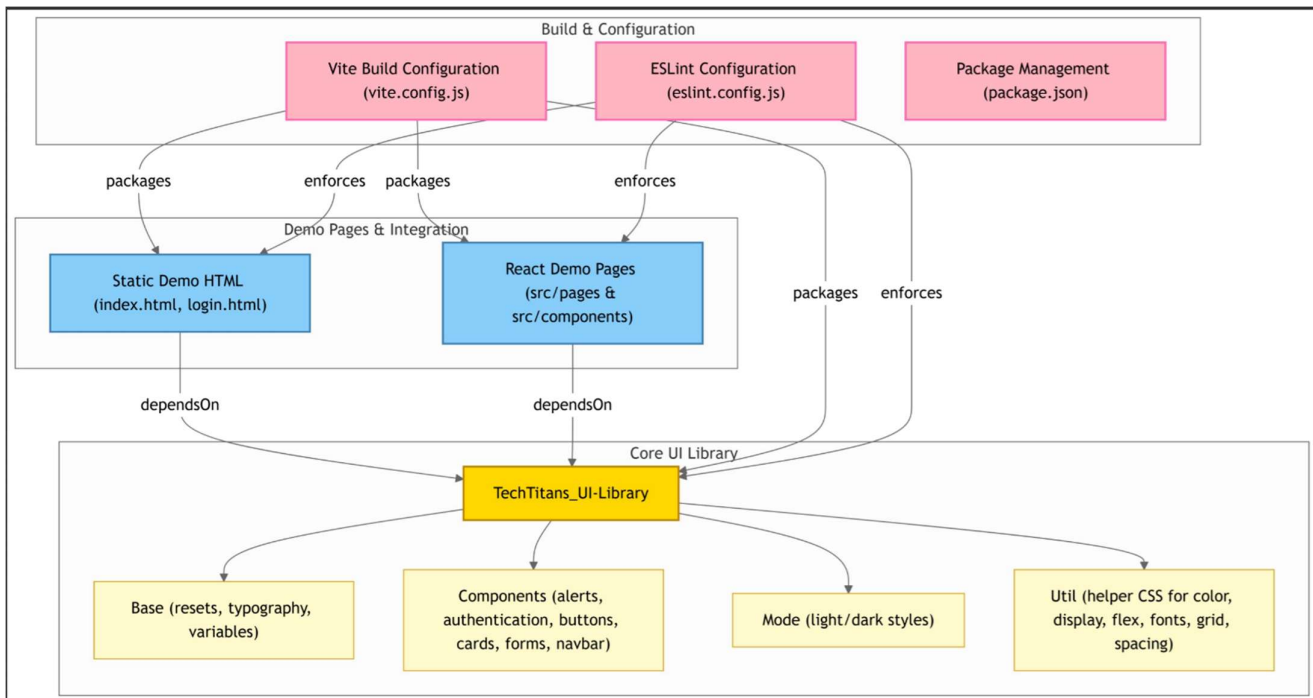
- Fully responsive 12-column grid system.
- Utility classes for spacing, typography, flex, and alignment.
- Built-in design system (color palette, spacing scale, typography).
- Semantic class naming for readability and maintainability.
- Reusable, modular components (buttons, cards, alerts, authentication , etc.).
- Live demo-ready HTML layout for testing.

## Technical Strengths:

- **Preconfigured** : No built in tools required , just drop it in and starting developing.
- **Developer Friendly** : Clean documentation , code available for each utility classes.
- **Optimised CSS** : Minimal footprints for faster loading.

# System Architecture

## Workflow :



## Component Structure:

```
src/
├── base/
│   ├── resetAll.css # Base styles and global settings
│   ├── type.css     # CSS file for resetting styles
│   ├── var.css      # CSS file for typography settings
│   └── var.css      # CSS file for defining variables
├── components/
│   ├── alerts.css   # UI components for the library
│   ├── authentication.css # CSS file for alert components
│   ├── button.css   # CSS file for authentication components
│   ├── card.css     # CSS file for button components
│   ├── form.css     # CSS file for card components
│   └── navbar.css   # CSS file for form components
├── modes/
│   └── light-dark.css # Theme or mode-specific styles
├── utils/
│   ├── color.css    # CSS file for light and dark mode styles
│   ├── display.css  # Utility styles and helper files
│   ├── flexing.css  # CSS file for color utilities
│   ├── font.css     # CSS file for display utilities
│   ├── grid.css     # CSS file for flexbox utilities
│   ├── space.css    # CSS file for font utilities
│   └── space.css    # CSS file for grid utilities
├── styles/
│   └── style.css     # CSS file for spacing utilities
└── style.css        # Global and main styles
                    # Main CSS styling file
```

# Tech Stack

- Core Language : CSS
- Build Tools : Vite
- Demo Sites : React , HTML/CSS
- Developer Tools : VS Code , GitHub
- Others : Figma , Canva

## Implementation Plan

### Timeline , Deliverables and Feature Rollout:

Work	Deliverables	<i>Feature Rollout</i>
11:00am to 1:00pm	<ul style="list-style-type: none"><li>• Finalize idea , repo setup , folder structure , role division</li></ul>	
1:00pm to 4:00pm	<ul style="list-style-type: none"><li>• Build Base Layer : resetAll.css , var.css , type.css</li><li>• Start utilities : spacing , flex , grid utilities</li></ul>	Base styles and variables
4:00pm to 8:00pm	<ul style="list-style-type: none"><li>• Core components : buttons.css , cards.css , alerts.css ,</li><li>• Apply semantics , readable class naming</li></ul>	Core utility classes and base components
8:00pm to 11:00pm	<ul style="list-style-type: none"><li>• Implement light-dark.css , CSS variables for modes</li><li>• Toggle based theme switching</li></ul>	Light/Dark Mode
11:00pm to	<ul style="list-style-type: none"><li>• Built Advanced Components</li></ul>	Advanced

3:00am	: navbar.css , form.css , authentication.css • Finalize utilities : color , display , font	components and extended utilities
3:00am to 7:00am	• Integrate into style.css • Documentation : class usage , examples	Final file styling and developer friendly docs
7:00am to 9:00am	• Build demo page • Refine styles and interactivity	Visual Presentation
9:00am to 11:00am	• Final testing : responsiveness , light/dark mode checks • Preparation of demo pitch and project submission	Submission

## Team Wise Responsibility Mapping:

1. Suraj Sharma : Coordinated team activities , Viewed project planning and timeline ,Developed core CSS utilities and components , Conducted testing for components.
2. Bhavesh Tripathi : Designed component layouts and user interfaces ,Optimized code for performance .
3. Sourabh Singh :Designed components layout, Developed core CSS utilities and components, Created design mockups .
4. Kartikey Wariyal : Designed component layouts and user interfaces , Implemented theming and dark mode features.

# Reusability and Documentation

SimpleUI fosters reusability by providing modular components and utility classes.

- **Modular Design** : Each and every component is styled separately allowing users to include what they need.
- **Utility Classes** : Classes like flex-row, space-between, and p-2 helped us in building layouts without writing custom CSS.

## Developer Documentation

A complete documentation would be soon available to assist the users and developers to with :

- **Getting Started**: Instructions on installation and setup.
- **Component Usage**: Examples and guidelines for each component.
- **Customization**: Details on modifying variables and themes.
- **Best Practices**: Recommendations for structuring and maintaining projects.

# References and Early work

## GitHub Repo Link :

[https://github.com/Suraj-Sharma001/SIMPLEUI-Tech\\_Titans.git](https://github.com/Suraj-Sharma001/SIMPLEUI-Tech_Titans.git)

## Release of first version of SimpleUI: Version v1.0.0

[https://github.com/Suraj-Sharma001/SIMPLEUI-Tech\\_Titans/releases/tag/v1.0.0](https://github.com/Suraj-Sharma001/SIMPLEUI-Tech_Titans/releases/tag/v1.0.0)

## Code Snippets:

```
var.css x
src > TechTitans_UI-Library > src > base > var.css > ...
1  :root {
2    --gui-primary: #4f46e5;
3    --gui-secondary: #64788b;
4    --gui-success: rgb(32, 128, 32);
5    --gui-danger: red;
6    --gui-warning: yellow;
7
8    --gui-font-sm: 0.875rem;
9    --gui-font-md: 1rem;
10   --gui-font-lg: 1.25rem;
11
12   --gui-radius: 0.5rem;
13   --gui-spacing-sm: 0.5rem;
14   --gui-spacing-md: 1rem;
15   --gui-spacing-lg: 1.5rem;
16   --gui-spacing-xl: 2.4rem;
17 }
18
```

```
style.css x
src > TechTitans_UI-Library > src > style.css
1  @import './base/resetAll.css';
2  @import './base/type.css';
3  @import './base/var.css';
4
5  @import './components/button.css';
6  @import './components/card.css';
7  @import './components/form.css';
8  @import './components/navbar.css';
9  @import './components/alerts.css';
10 @import './components/form.css';
11 @import './components/authentication.css';
12
13 @import './mode/light-dark.css';
14
15 @import './util/color.css';
16 @import './util/display.css';
17 @import './util/flexing.css';
18 @import './util/grid.css';
19 @import './util/space.css';
20 @import './util/font.css';
```



```
App.jsx x
src > App.jsx > ...
1 import { Routes, Route } from 'react-router-dom';
2 import Layout from '../components/Layout';
3 import Home from '../pages/Home';
4 import Button from '../pages/Button';
5 import Card from '../pages/Card';
6 import Alert from '../pages/Alert';
7 import Auth from '../pages/Auth';
8 import Utilities from '../pages/Utilities';
9
10 function App() {
11   return (
12     <Layout>
13       <Routes>
14         <Route path="/" element={<Home />} />
15         <Route path="/buttons" element={<Button />} />
16         <Route path="/cards" element={<Card />} />
17         <Route path="/alerts" element={<Alert />} />
18         <Route path="/auth" element={<Auth />} />
19         <Route path="/utilities" element={<Utilities />} />
20       </Routes>
21     </Layout>
22   );
23 }
24
25 export default App;
26
```

## Components Preview:

[Home](#) [Buttons](#) [Cards](#) [Alerts](#) [Auth](#) [Utilities](#) 

### Auth Forms

Login and Sign Up form templates.

Login

Login

[Forgot Password?](#) [Sign Up](#)

```
<div class="gui-auth">
  <h2 class="gui-auth-header">Login</h2>
  <form class="gui-auth-form">
    <input type="email" class="gui-input" placeholder="Email" />
    <input type="password" class="gui-input mt-2" placeholder="Password" />
    <button class="gui-btn gui-btn-success mt-2 w-full">Login</button>
  </form>
  <div class="gui-auth-footer">
    <a href="#">Forgot Password?</a>
    <a href="#">Sign Up</a>
  </div>
</div>
```

## Alerts

GUI alert components for messages.

Success! All good.

Warning! Are you Sure??.

Error! Something went wrong.

```
<div class="gui-alert gui-alert-success">Success! All good.</div>
<div class="gui-alert gui-alert-warning">Warning! Are you Sure??.</div>
<div class="gui-alert gui-alert-danger">Error! Something went wrong.</div>
```

## Buttons

GUI button styles for common actions.

Primary

Secondary

Success

```
<button class="gui-btn gui-btn-primary">Primary</button>
<button class="gui-btn gui-btn-secondary">Secondary</button>
<button class="gui-btn gui-btn-success">Success</button>
```

## Justify Centre

Centered Text

```
<div class="flex-row text-pink justify-centre gui-border p-4 flex">
  <span>Centered Text</span>
</div>
```

## Typography

Small and Light Text

Normal Text

Large and Semibold Text

Extra large and Bold Text

```
<p class="gui-text-sm gui-font-light">Small and Light Text</p>
<p class="gui-text gui-font-normal">Normal Text</p>
<p class="gui-text-lg gui-font-semibold">Large and Semibold Text</p>
<p class="gui-text-xl gui-font-black">Extra large and Bold Text</p>
```

# References:

<https://vite.dev/guide/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/en-US/docs/Web>