CoreCompass – Multi-Utility Web Application
Project Report

Developed By:

• BALAJI [212223040023] - CSE
• RENUKA [212223240136] - AIML

Abstract

CoreCompass is a responsive, multi-utility web application developed using HTML, CSS, and JavaScript. It integrates BMI calculation, calorie tracking, country information, emergency numbers, weather updates, and a translation module. The system aims to provide a unified, lightweight browser-based solution requiring no installation. API integrations ensure real-time data accuracy with a smooth, responsive interface.

1. Introduction

CoreCompass addresses the need for consolidated digital tools. Instead of using multiple separate apps for fitness, weather, translation, and country information, users can access all features in one unified web interface. The application is fully frontend-based, ensuring broad compatibility and ease of deployment.

2. Objectives

• Build a multi-feature web app using HTML, CSS, and JavaScript.
• Provide BMI and calorie calculation tools.
• Offer country-level information including emergency numbers.
• Integrate real-time weather data.
• Provide translation between multiple languages.
• Maintain a clean, responsive user interface.

3. Problem Statement

Users often rely on multiple platforms to perform basic tasks like health checking, weather lookup, or translation. This leads to inefficiency and

time waste. CoreCompass solves this by merging essential utilities into one web-based system.

4. Proposed System

The CoreCompass system provides:

1. Fitness Tools:
    - BMI Calculator
    - Calorie Counter

2. Country Module:
    - Map, Capital, Language
    - President / Leader
    - Emergency Numbers

3. Weather Module:
    - City-wise weather updates

4. Translation Module:
    - Text-to-text translation via API

5. Tools and Technologies Used

- HTML5, CSS3, JavaScript
- REST APIs (Country, Weather, Translation)
- Visual Studio Code
- Font Awesome, Google Fonts
- Hosting: GitHub Pages / Netlify (optional)

6. System Design

The system uses a client-side architecture where user inputs are processed using JavaScript, and additional data is retrieved from public APIs. The design follows tab-based navigation with modular separation.

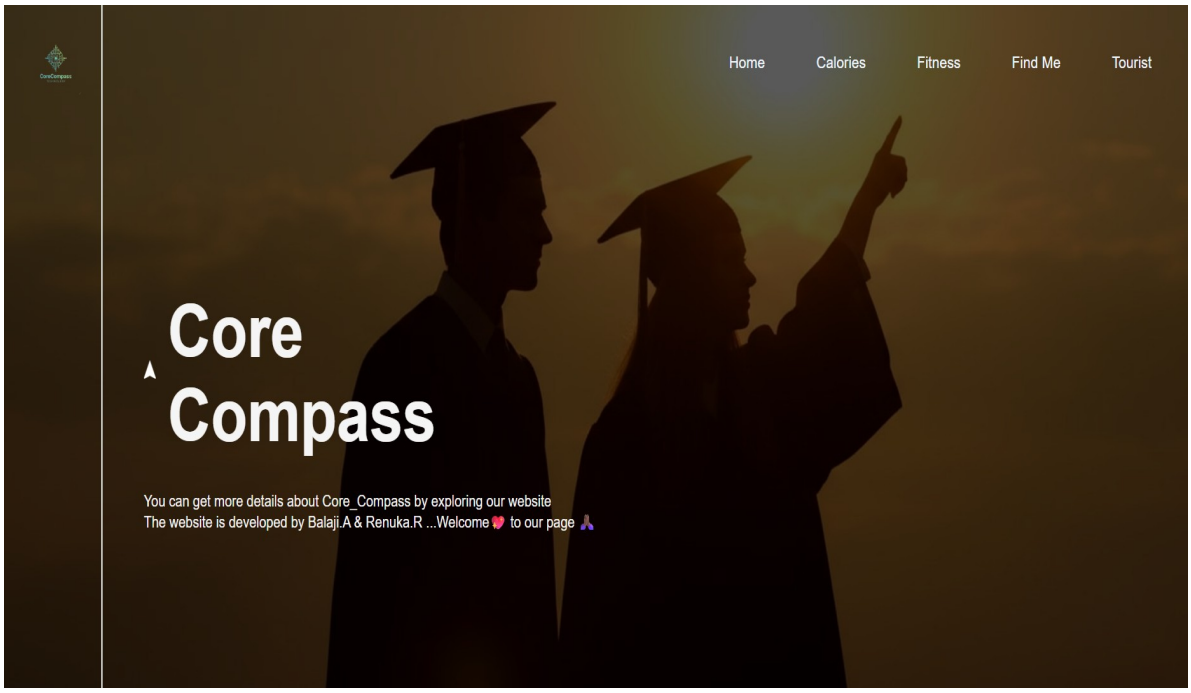
7. Implementation

- Fitness tools compute BMI and calorie estimates.
- Country module fetches geographic and administrative information.
- Weather module retrieves meteorological data.
- Translator uses a language API to convert text.
- CSS ensures responsive layout.

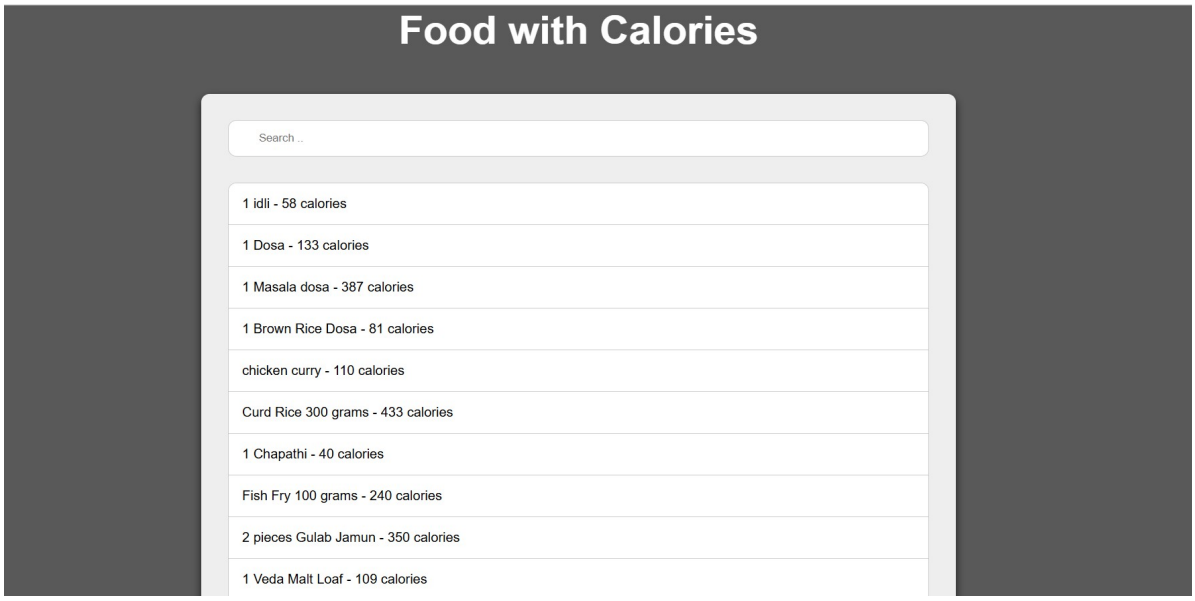
8. JavaScript Features

- Fetch API integration for data retrieval
- Input validation and error alerts
- API error fallback handling
- DOM manipulation for dynamic UI updates

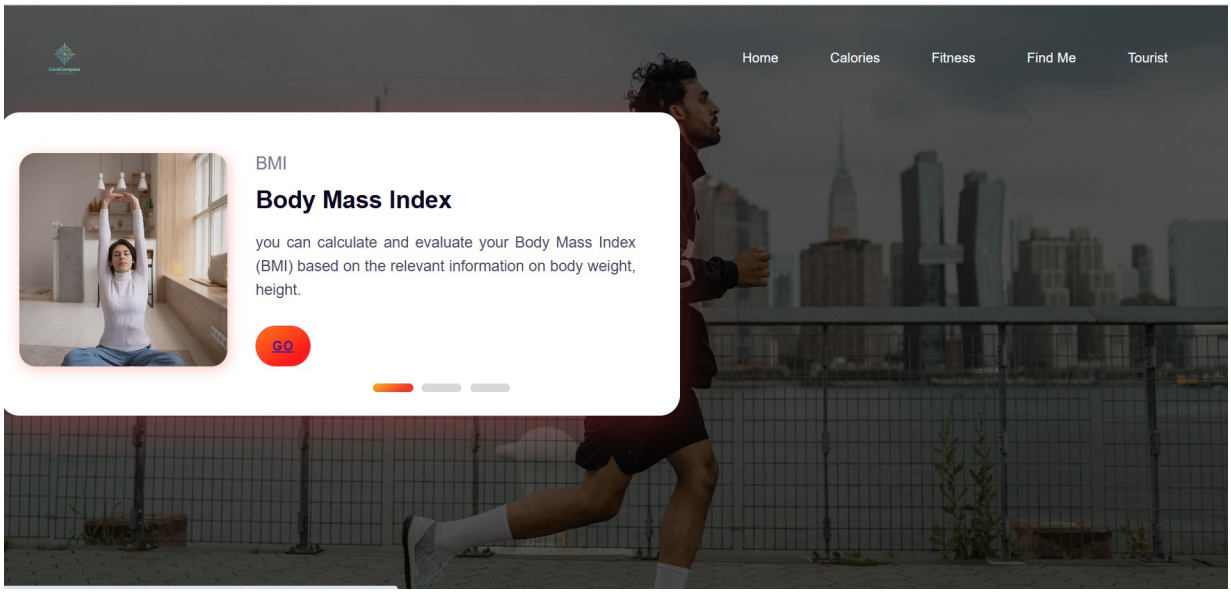
Home Page



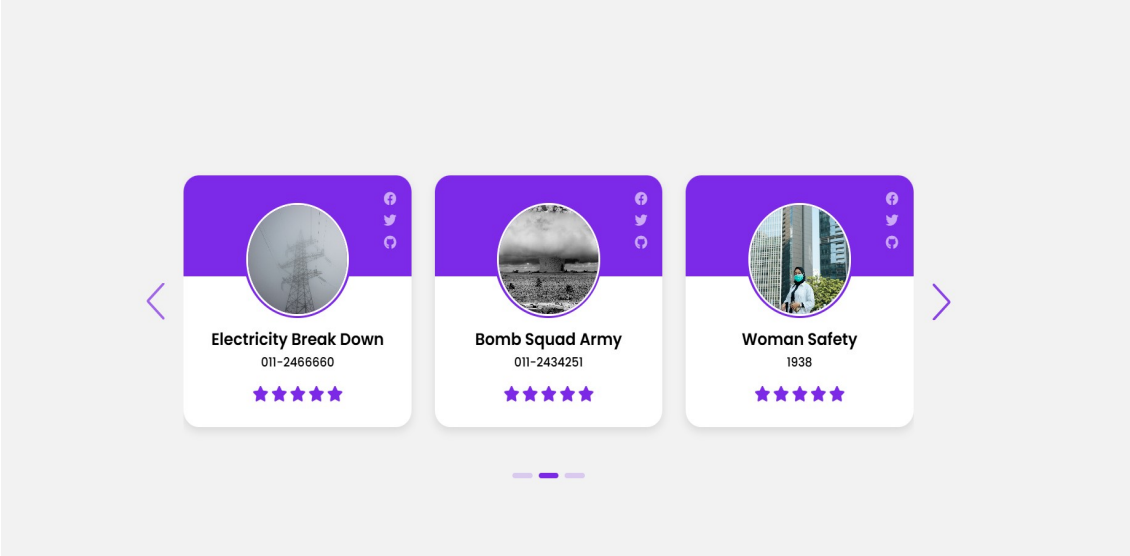
Calories page



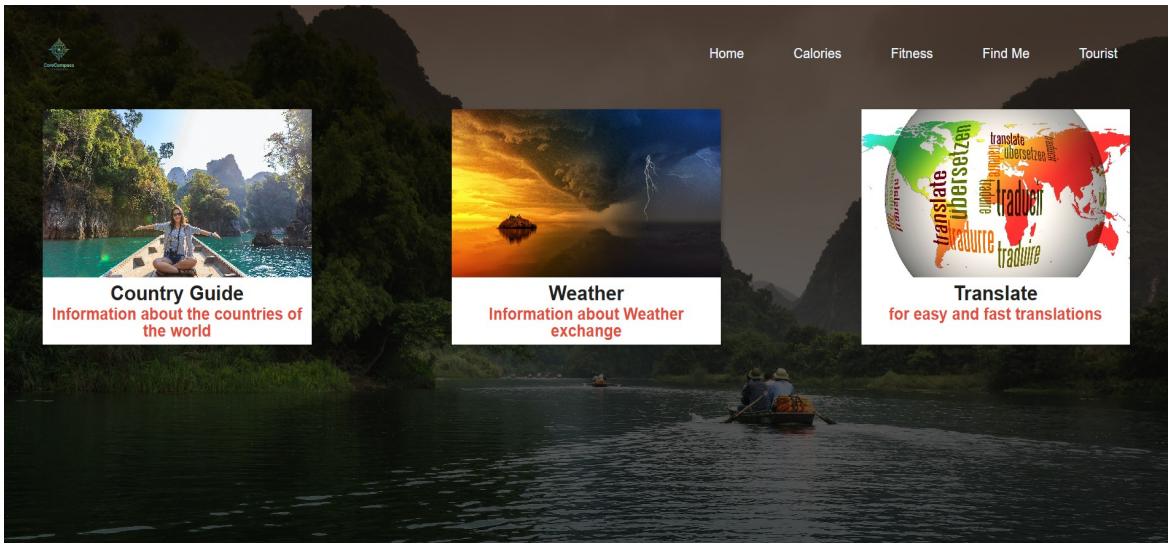
BMI Calculator



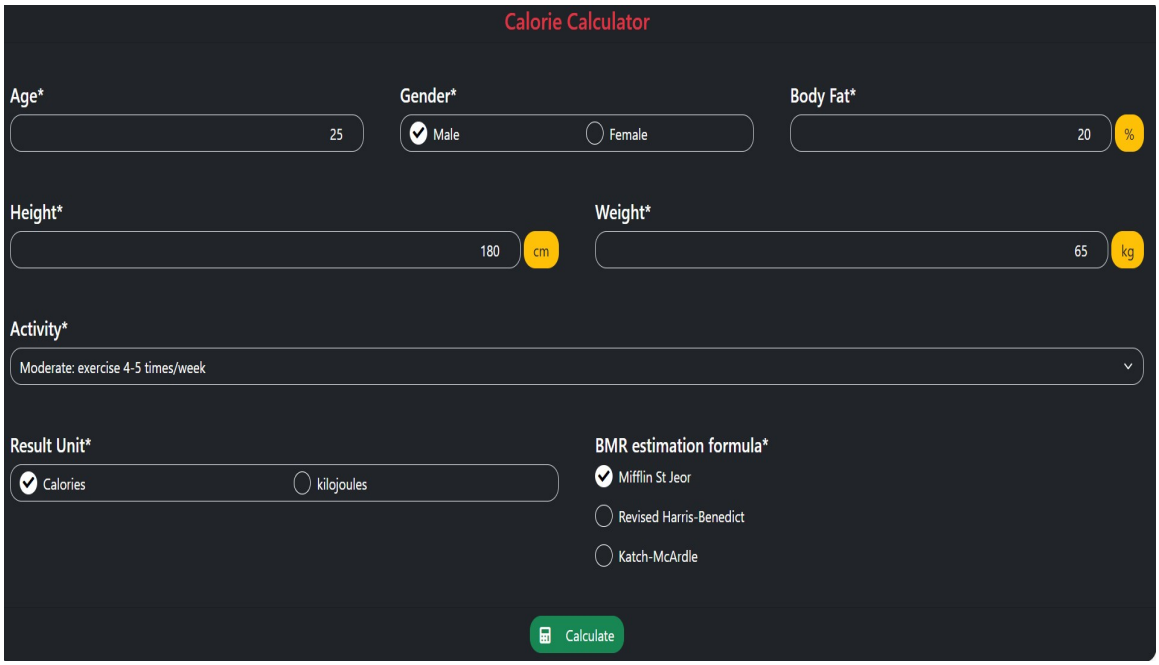
Emergency Page



Tourist Page



Calories Calculator



9. Testing

The following table summarizes major test cases performed on the system:

Test Case	Input	Expected Output	Result
BMI Invalid Input	Height= '', Weight=''	Show error message	Pass
Country Not Found	Country = 'xyzabc'	Display 'Country Not Found'	Pass
Valid Weather Search	City = 'Chennai'	Show weather details	Pass

Translate Text	English → Tamil	Return translated output	Pass
Calorie Counter Invalid	Activity='' or invalid	Show validation message	Pass

10. Results and Discussion

The application performs smoothly across different browsers and devices. All modules respond accurately and handle user errors properly. API-based features deliver real-time information efficiently.

11. Conclusion and Future Enhancements

CoreCompass, developed by Balaji and Renuka, demonstrates an effective multi-utility web platform. Future improvements may include dark mode, voice input, offline support, saved preferences, and enhanced UI themes.

12. References

- https://www.w3schools.com
- https://developer.mozilla.org
- https://fonts.google.com