**Software Requirements**

**Specification**

**for**

**EcoCredit**

**Prepared by Harsh Garg, Sai Mahesh, Pushkar Bhat, Shourya Pratap, Aman**

**3rd September 2023**

# Introduction

The Carbon Credits Gamification Application is an innovative software solution designed to revolutionize and gamify the pursuit of environmental sustainability within Shell and beyond. In alignment with Shell's commitment to achieving netzero emissions, this application seeks to engage and empower Shell employees by incentivizing and rewarding carbon footprint reduction activities.

Through a combination of cutting-edge technology and gamification principles, this platform aims to inspire individuals to adopt eco-conscious behaviours, foster a culture of environmental responsibility, and contribute to the global effort to combat climate change.

This Software Requirement Specification document outlines the comprehensive features, functionalities, and technical requirements that will drive the successful development and deployment of this transformative application.

# Abstract

The Carbon Credits Gamification Application is a web and mobile-based platform designed to motivate and reward Shell employees for adopting ecofriendly practices and reducing their carbon footprint. By engaging users in a gamified experience, the application aims to contribute to Shell's net-zero emissions initiative while promoting environmental consciousness.

# Challenges

1. Carbon Emissions Tracking: Implementing a reliable system to accurately track and calculate individual carbon emissions reductions, considering different activities and factors.
2. Activity Validation: Developing mechanisms to validate user-reported activities to prevent fraudulent or inaccurate claims.
3. Gamification Mechanics: Designing engaging gamification elements that encourage long-term user participation.
4. Data Security and Privacy: Ensuring the security of user data and compliance with data protection regulations.
5. Scalability: Preparing the application to handle a potentially large number of users efficiently.

# Solution

The Carbon Credits Gamification Application will address these challenges through the following strategies:

1. Carbon Emissions Tracking: Implement a data collection system that integrates with external sources for accurate emissions data. Utilize PostgreSQL to store and manage historical emission data.
2. Activity Validation: Implementing methods to verify user-submitted data, such as checking against third-party sources or requiring photo evidence, can help ensure the accuracy of reported activities and the fairness of the Carbon Credits allocation process.
3. Gamification Mechanics: Employ React.js to create an engaging and intuitive user interface featuring badges, leaderboards, rewards, and notifications.
4. Data Security and Privacy: Implement robust security measures, including data encryption, secure authentication, and authorization, utilizing .NET Core's security features.
5. Scalability: Design the application architecture in .NET Core to be scalable, allowing for the seamless addition of users and activities.

# Scope

The Carbon Credits Gamification Application will encompass the following key features and functionalities:

1. User Management: Registration, authentication, profile management, and password reset.
2. Carbon Emissions Tracking: Data input for various emission-reducing activities, real-time carbon emissions calculation, and secure storage in PostgreSQL.
3. Carbon Credits Calculation: Calculation of carbon credits based on user activities, integrating with established emission reduction standards.
4. Gamification: React.js will be used to create an interactive user experience, including badges, leaderboards, rewards, and notifications.
5. Reporting and Analytics: User-friendly dashboards for users to visualize their carbon footprint and credits, historical data tracking, and insightful analytics.
6. Integration: Integration with external data sources for emission data and potentially with Shell's existing systems.

# Functional Requirements

1. User Management
   1. User Registration: Users can sign up with their Shell credentials, providing essential information.
   2. Authentication and Authorization: Secure login and access control.
   3. Profile Management: Users can update their profiles and preferences.
   4. Password Reset: Users can request a password reset link.
2. Carbon Emissions Tracking
   1. Data Input: Users can log various activities that reduce carbon emissions.
   2. Real-time Calculation: The application calculates carbon emissions reduction in real-time.
   3. Data Storage: Store historical emission data in the PostgreSQL database.
3. Carbon Credits Calculation
   1. Calculation Algorithm: Implement a robust algorithm to calculate carbon credits based on activity data.
   2. Integration: Integrate with established emission reduction standards and formulas.
4. Gamification
   1. Badges: Users can earn badges for achieving specific milestones.
   2. Leader boards: Encourage competition by displaying rankings based on carbon credits earned.
   3. Reward System: Reward users with carbon credits for achieving specific goals or completing challenges.
   4. Notifications: Send notifications to engage users and inform them of updates and achievements.
5. Reporting and Analytics
   1. Dashboard: Display carbon footprint and credits on a user-friendly dashboard.
   2. Historical Data: Enable users to view their historical emissions data and track progress.
   3. Insights: Provide insights into how specific activities impact carbon emissions.

# Non-Functional Requirements

1. Security: Implement data encryption, secure authentication, and authorization using .NET Core's security features.
2. Scalability: Design the application architecture in .NET Core to be scalable, allowing for horizontal scaling of servers and load balancing.
3. Performance: Optimize React.js for fast and responsive user interfaces and implement efficient data retrieval from PostgreSQL.
4. Privacy: Ensure compliance with data protection regulations, including GDPR.
5. Availability: Minimize downtime through load balancing and redundancy.
6. Compatibility: Support multiple web browsers and mobile devices with a responsive design.

# Users

The Carbon Credits Gamification Application defines two primary user roles:

1. Standard User: Normal users are Shell employees who have registered and use the application to track their carbon footprint reduction efforts, participate in gamified activities, earn Carbon Credits, and engage with sustainability initiatives.
2. Admin User: Admin users oversee and manage the Carbon Credits Gamification Application. They have additional privileges related to system administration, user management, activity management, and ensuring the application's overall functionality.

These two distinct user roles ensure that the Carbon Credits Gamification Application remains user-friendly and secure, empowering normal users to actively participate in sustainability efforts while providing administrators with the tools necessary for effective system management and oversight.

### Use Case Diagram

[Include a use case diagram illustrating various user interactions with the system, including registration, emissions tracking, rewards, etc.]

### Entity-Relationship (ER) Diagram

[Provide an ER diagram depicting the database structure, including tables for users, emissions data, activities, and rewards.]

### Sequence Diagram

[Create sequence diagrams to illustrate interactions between system components, such as user registration, emissions tracking, and reward allocation.]

---

This expanded SRS document provides a more detailed overview of your project, taking into account your chosen tech stack and addressing the challenges and solutions for each aspect of the application. You can now use this as a blueprint for the development of your Carbon Credits Gamification Application.