A

SYNOPSIS

of

MINOR PROJECT

on

Credit Card Fraud Detection

Submitted by

Shashwat Sharma

(21EGICS107)

Project Guide
Ms. Ruchi Vyas

Head of Department
Dr. Mayank Patel

Geetanjali Institute of Technical Studies, Dabok, Udaipur (Raj.)
Department of Computer Science and Engineering
June, 2024

PROBLEM STATEMENT

- In today's digital economy, credit card transactions have become ubiquitous. However, the rise in credit card usage has also led to an increase in fraudulent activities.

- One of the fundamental steps in mitigating credit card fraud is to ensure that credit card numbers are valid before processing transactions.

- This project addresses this issue by developing a web-based application that validates credit card numbers using the Luhn algorithm.

- This algorithm helps determine whether a given credit card number is syntactically valid, thus providing a preliminary check before more sophisticated anti-fraud measures are employed.

BRIEF DESCRIPTION

- This project involves the creation of a web application designed to validate credit card numbers entered by users.

- The validation process employs the Luhn algorithm, a widely-used checksum formula that helps detect errors in credit card numbers.

- The application consists of a user-friendly frontend built with HTML, CSS, and JavaScript, and a backend implemented using Flask, a Python web framework.

- Users can input their credit card numbers into a form on the webpage, and the application will immediately inform them whether the number is valid or invalid.

OBJECTIVE & SCOPE

Objective:

The primary objective of this project is to create a reliable and efficient tool for validating credit card numbers. Specific objectives include:

- Developing an intuitive and accessible user interface for entering credit card numbers.
- Implementing the Luhn algorithm in Python to accurately validate the entered credit card numbers.
- Ensuring real-time feedback for users regarding the validity of their credit card numbers.

Scope:

The scope of this project includes:

- Building a web-based application accessible from any device with an internet connection.
- Providing basic validation of credit card numbers using the Luhn algorithm.
- Offering a foundation that can be extended with additional features, such as identifying the type of credit card (Visa, MasterCard, etc.) and integrating with other security measures.

METHODOLOGY

The project development is divided into several phases:

1. Frontend Development:

- **Design**: A clean and minimalistic design is chosen to ensure the application is easy to use. The form where users enter their credit card number is centrally placed for easy access.

- **Implementation**: HTML is used to create the structure of the web page, CSS for styling to make the page visually appealing and responsive, and JavaScript for handling user interactions and form submissions.

2. Backend Development:

- **Framework**: Flask, a lightweight Python web framework, is used to build the server-side application. Flask is chosen for its simplicity and flexibility, making it ideal for this project.

- **Algorithm Implementation**: The Luhn algorithm is implemented in Python. This algorithm calculates whether a given credit card number is valid by performing a series of arithmetic operations.

3. Integration:

- **AJAX Requests**: JavaScript is used to handle form submissions without reloading the page. When a user submits a credit card number, an AJAX request sends this data to the Flask backend.

- **Response Handling**: The backend processes the credit card number using the Luhn algorithm and returns a JSON response indicating whether the number is valid. This response is then displayed on the webpage.

HARDWARE & SOFTWARE REQUIREMENTS

Hardware:

- A personal computer or laptop with sufficient processing power and memory to run Python and Flask.

- Internet access for development, testing, and deployment purposes.

Software:

- **Programming Language**: Python 3.x

- **Web Framework**: Flask

- **Frontend Technologies**: HTML5, CSS3, JavaScript

- **Development Tools**: A code editor or IDE (e.g., Visual Studio Code, PyCharm), a web browser for testing (e.g., Google Chrome, Mozilla Firefox)

- **Version Control**: Git for tracking changes and collaborating (optional but recommended)

TECHNIQUES

- **Frontend**:
    - **HTML**: Provides the structure for the web page.
    - **CSS**: Used for styling and making the page responsive.
    - **JavaScript**: Handles form submissions and communicates with the backend via AJAX requests.

- **Backend**:
    - **Python**: The primary programming language used for implementing the backend logic and the Luhn algorithm.
    - **Flask**: A lightweight web framework for Python, used to create the server and handle requests.

- **Validation Algorithm**: The Luhn algorithm, a simple checksum formula used to validate various identification numbers, primarily credit card numbers.

TESTING TECHNIQUES

Unit Testing:

Unit tests are created for the Luhn algorithm to ensure it can accurately determine the validity of various credit card numbers. These tests include:

- **Valid Credit Card Numbers**: Testing known valid credit card numbers to confirm they pass the validation.
- **Invalid Credit Card Numbers**: Testing known invalid credit card numbers to ensure they fail the validation.

Integration Testing:

Integration tests ensure that the frontend and backend work together seamlessly. These tests involve:

- Submitting a valid credit card number through the form and verifying the correct validation message is displayed.
- Submitting an invalid credit card number and checking that the appropriate error message is shown.

User Testing:

The application is tested by a group of users to gather feedback on usability and functionality. Users are asked to:

- Enter various credit card numbers and observe the validation results.
- Provide feedback on the interface design, ease of use, and any issues encountered.

PROJECT CONTRIBUTION

This project makes several contributions:

- **Practical Implementation**: Provides a practical example of using the Luhn algorithm for credit card validation, demonstrating how it can be integrated into a web application.

- **Educational Value**: Serves as an educational tool for understanding web development with Flask and implementing basic validation algorithms.

- **Foundation for Further Development**: Offers a foundation that can be expanded with additional features, such as identifying card types, adding security measures, and integrating with larger payment processing systems.