f

FUN FACT GENERATOR

MICRO PROJECT REPORT

Submitted by

JEENAS DEEPA (23ITR070)

JEEVASHRE (23ITR071)

NANDHINI (23ITR108)

in partial fulfilment of the requirements

for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

NOVEMBER 2024

DEPARTMENT OF INFORMATION TECHNOLOGY

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638060

NOVEMBER 2024

BONAFIED CERTIFICATE

This is to certify that the Project report entitled **FUN FACT GENERATOR** is the bonafide record of project work done by **JEENAS DEEPA A (23ITR070), JEEVASHRE G (23ITR071), NANDHINI E (23ITR108)** for **22ITT32 PYTHON PROGRAMMING AND FRAMEWORKS** during the year 2024–2025.

COURSE IN CHARGE **HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:

Submitted for the final viva voce examination held on_____.

DEPARTMENT OF INFORMATION TECHNOLOGY

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638060

NOVEMBER 2024

DECLARATION

We affirm that the Project Report titled **FUN FACT GENERATOR** being submitted in partial fulfilment of the requirements for the award of Bachelor of Technology is the original work carried out by us. It has not formed the part of anyother project report or dissertation on the basis of which a degree or award was conferred on anearlier occasion on this or any other candidate.

Date:

JEENAS DEEPA A
23ITR070

JEEVASHRE G
23ITR071

NANDHINI E
23ITR108

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date:

Name and Signature of the course in charge

ABSTRACT

The Fun Fact Generator web application is an innovative project aimed at providing users with a platform to discover and engage with interesting, random facts across diverse categories such as Science, History, and Technology. This application combines education and entertainment to enhance user knowledge in a captivating and interactive way. The central goal of the project is to make learning enjoyable by presenting bite-sized information that is both fun and memorable. The web application is designed with user-centric features, including an aesthetically pleasing homepage that serves as the main hub for discovering facts. Users are greeted with a featured random fact and have the option to generate new ones at the click of a button. Advanced filtering mechanisms allow users to customize their fact feed based on personal interests or specific themes, ensuring a tailored and engaging experience. Additionally, user authentication features, such as login and signup pages, provide a secure and personalized environment for accessing the app.

The backend of the application leverages the Python programming language, supported by the Flask web framework for efficient handling of server-side logic. MySQL is employed for robust database management, ensuring reliable storage and retrieval of facts and user data. On the frontend, the application uses HTML to structure content, CSS for styling and layout, and JavaScript to enable dynamic interactions and real-time user engagement. To ensure smooth functionality and future scalability, the development process has adhered to modular principles. The Fun Fact Generator module facilitates the core functionality of delivering random facts, while the authentication module ensures secure user access. Potential future enhancements include features like user-submitted facts, personalized preferences for categories, and a broader range of topics to explore. By integrating advanced web technologies and user-friendly design principles, this project demonstrates how a simple idea can be transformed into a powerful educational tool. The Fun Fact Generator not only entertains users but also contributes to their knowledge in a light-hearted and enjoyable manner, bridging the gap between learning and leisure.

ACKNOWLEDGEMENT

First and foremost, we acknowledge the abundant grace and presence of Almighty throughout different phases of the project and its successful completion.

We wish to express our gratefulness to our beloved Correspondent **Thiru.A.K.ILANGO B.Com., M.B.A., LLB.,** and all the trust members of Kongu VellalarInstitute of Technology Trust for providing all the necessary facilities to complete the projectsuccessfully.

We express our deep sense of gratitude to our beloved Principal **Dr.V.BALUSAMY B.E.(Hons)., M.Tech., Ph.D.,** for providing us an opportunity tocomplete the project.

We express our gratitude to **Dr. S. ANANDAMURUGAN M.E., Ph.D.,** Head of the Department, Department of Information Technology for his valuable suggestions.

We are highly indebted to **Mrs. P. VANITHA B.E., M.E.,** Department of Information Technology for her valuable supervision and advice for the fruitful completion of the project.

We are thankful to the faculty members of the Department of Information Technology for their valuable guidance and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CSS	Cascading Style Sheet
HTML	HyperText Markup Language
ID	Identification
JS	JavaScript
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Fun Fact Generator Web Application is a unique platform designed to provide users with a delightful and educational experience by offering random, intriguing facts across multiple categories, including Science, History, and Technology. The core idea is to combine entertainment with learning, making the process enjoyable and memorable. In today's digital era, users seek engaging and quick ways to gain knowledge, and this application fulfills that need with its simple, user-friendly interface. The application stands out by offering personalization features, such as filtering facts by specific themes or categories, ensuring that users can explore content tailored to their interests. The seamless integration of backend and frontend technologies ensures a smooth and interactive user experience, while user authentication mechanisms provide a secure environment. This project demonstrates the effective use of modern web development technologies, including Python for backend logic, Flask as the web framework, MySQL for database management, and HTML, CSS, and JavaScript for creating an interactive user interface. It highlights the potential of web applications in promoting informal education and entertainment simultaneously.

1.2 OBJECTIVE

The primary objective of the Fun Fact Generator Web Application is to create an engaging platform where users can explore random facts in a personalized and interactive manner. Allow users to instantly access random, interesting facts with a single click, promoting quick learning. Include a wide variety of topics, such as Science, History, and Technology, to cater to diverse user interests. Implement advanced filtering options, enabling users to tailor their fact feed according to their preferences or specific themes. Design a user-friendly interface for easy navigation and interaction, with secure login and signup features to enhance accessibility. Foster an informal and enjoyable learning experience that combines entertainment with knowledge acquisition.

CHAPTER 2

SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

Processor	:	Intel Core i3
Processor Speed	:	Max. of 3.50GHz
RAM	:	8GB RAM
Hard Disk	:	512GB
Keyboard	:	Standard 104 enhanced
Mouse	:	Local PS/2

2.2 SOFTWARE REQUIREMENTS

Platform	:	Visual Studio Code
Language	:	Python 3.12, Jinja2, HTML, CSS, Bootstrap, Javascript
Database	:	SQL
Library	:	SQL, Flask
Framework	**:**	Flask

2.3 SOFTWARE DESCRIPTION

2.3.1 Visual Studio Code

Visual Studio Code is a versatile and user-friendly code editor used by developers worldwide. It supports a wide range of programming languages and frameworks, including JavaScript, Jinja2, HTML and CSS. For this project, VS Code provides a convenient environment for writing, editing, and debugging code related to the Server side like Flask framework and flasksqlalchemy, as well as client-side scripting with Jinja2 for rendering HTML templates. Its built-in terminal and debugging tools streamline the development process, while its customizable features allow developers to tailor the editor to their specific needs, enhancing productivity and efficiency throughout the project lifecycle.

Python 3.12

Python 3.12 is the latest major release of the Python programming language, offering enhanced performance, better error messages, and new features to improve developer productivity. One of the key highlights of Python 3.12 is its improved performance, with optimizations in the interpreter that lead to faster execution of Python code. The release includes changes to the typing module, making type annotations simpler and more expressive, which helps in writing better-structured and maintainable code. Additionally, Python 3.12 introduces refined error messages, providing clearer explanations and pinpointing the exact location of issues, making debugging easier for developers. Enhanced features in the standard library and updates to popular modules, such as improvements in str and decimal, further enhance functionality and usability. This version also brings deprecations and removal of outdated features, encouraging developers to adopt modern, best practices. Python 3.12 continues the tradition of making Python a versatile, efficient, and developer-friendly language for a wide range of applications.

2.3.2 Jinja2

Jinja2 is a powerful and flexible templating engine for Python, commonly used with web frameworks like Flask and Django. It allows developers to separate the HTML structure from the Python backend logic, making it easier to create dynamic and interactive web

applications. Jinja2 uses a familiar syntax, similar to Python, with control structures such as loops and conditionals, enabling complex HTML generation.

The templating engine provides features like template inheritance, which allows developers to define a base template and extend it for different pages, promoting code reusability. It also includes robust filtering options and custom macros for transforming data before rendering it in the template. Jinja2's automatic escaping feature helps protect against Cross-Site Scripting (XSS) attacks, enhancing web application security

2.3.3 SQL

SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases. It provides a powerful and flexible way to interact with data stored in tables, making it possible to perform a variety of operations like querying, updating, inserting, and deleting data. SQL is essential for working with relational database management systems (RDBMS) such as MySQL, PostgreSQL, Oracle, and SQLite. SQL allows developers to define database schemas, create and modify tables, and enforce data integrity constraints, ensuring the accuracy and consistency of the data.

2.3.4 Flask

Flask is a lightweight and flexible web framework for Python that is designed to make web development simple and efficient. Known for its minimalistic approach, Flask provides the essential tools to build web applications without imposing unnecessary complexity. It follows the WSGI (Web Server Gateway Interface) standard and is often referred to as a "micro-framework" because it doesn't come with built-in libraries like authentication, database management, or form handling, giving developers the freedom to choose the components they need.

Flask is highly extensible, allowing developers to add additional features through extensions, such as database integration (e.g., Flask-SQLAlchemy), form handling (e.g., Flask-WTF), and user authentication (e.g., Flask-Login). The framework also uses Jinja2 for templating, enabling the dynamic generation of HTML content. Due to its simplicity and flexibility, Flask is widely used for developing small to medium-sized web applications.

CHAPTER 3

SYSTEM DESIGN

3.1 HOMEPAGE MODULE

Diagram Type: Use Case Diagram

User: Interacts with the homepage to view fun facts and navigate.

System: Handles dynamic fact retrieval and page rendering.

View Random Fact: Users can click a button to see random facts.

Navigate to Login/Signup: Users can move to the authentication pages.

Filter Facts: Optional functionality for selecting categories of fact.

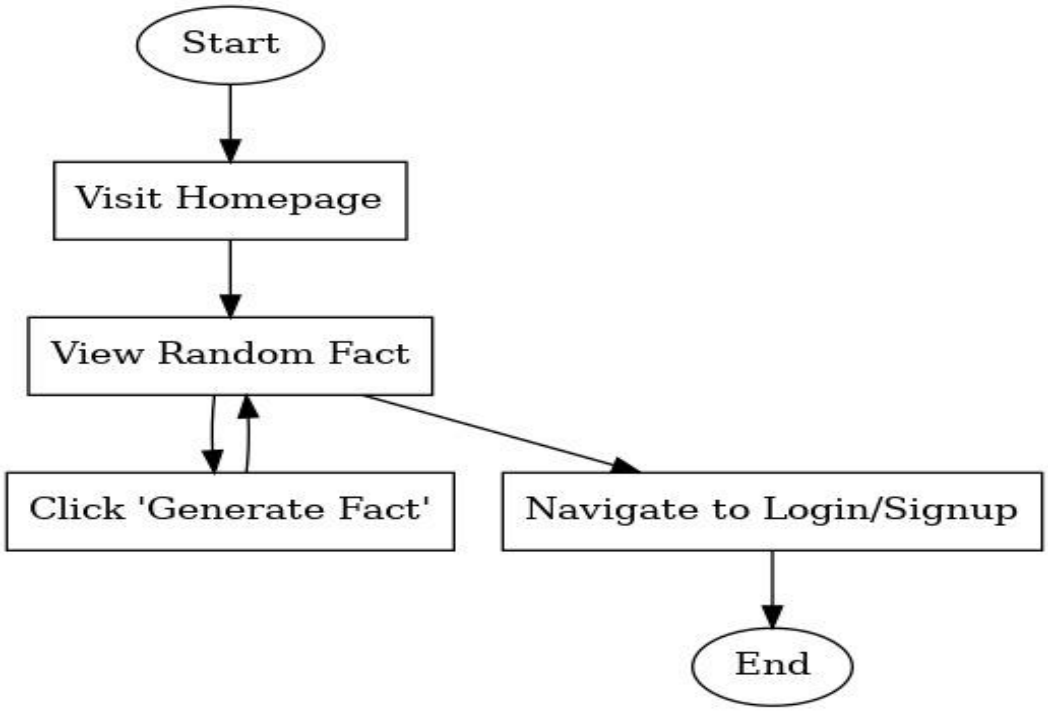

Fig 3.1 Use Case Diagram

3.2 FUN FACT GENERATOR MODULE

Diagram Type: Sequence Diagram

Scenario: Generating a fun fact

User clicks the "Generate Fact" button.

SystemManager sends a request to the Database.

Database retrieves a random fact and returns it to the SystemManager.

SystemManager displays the fact on the Homepage.

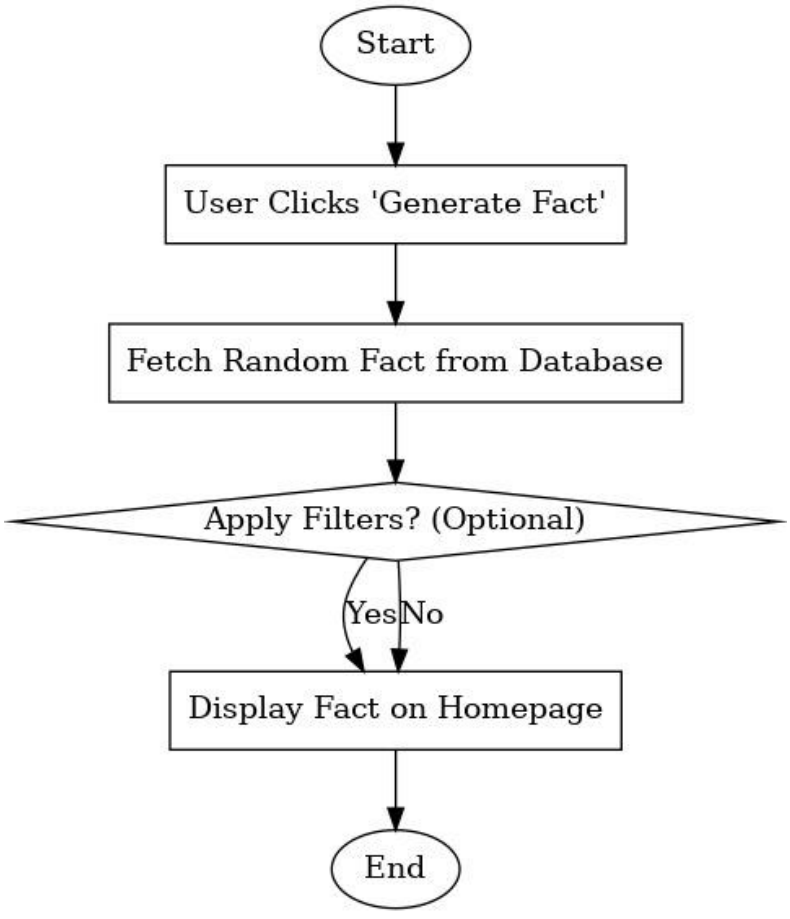

Fig 3.2 Class Diagram

3.3 LOGIN AND SIGNUP MODULE

Diagram Type: Activity Diagram

Start: User accesses the login/signup page.

Decision Point: Check if the user is already registered.

Yes: Redirect to the login form.

No: Redirect to the signup form.

Login Flow:

Input credentials → Validate credentials in the Database.

If valid, allow access; else, prompt an error message.

Signup Flow:

Input user details → Store details in the Database.

Redirect to the login page upon successful signup.

End.

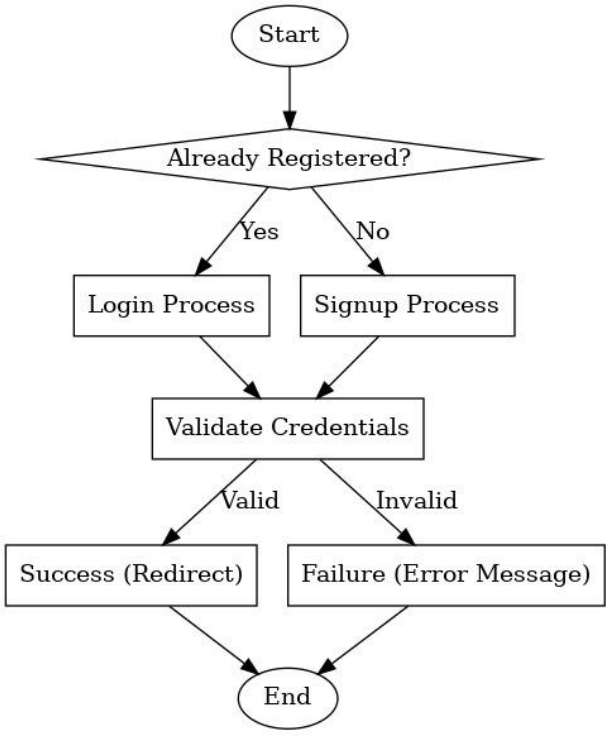

Fig 3.3 Sequence Diagram

CHAPTER 4

Modules Description

4.1 Fun Fact Generator Implementation

The Fun Fact Generator web application aims to deliver random, interesting facts to users at the click of a button. Users can explore facts across various categories such as Science, History, and Technology. The application includes advanced filtering options, enabling users to tailor their fact feed according to their preferences or specific themes. The homepage features an inviting layout showcasing a random fun fact prominently, with options to generate new facts easily. Future enhancements may include the ability for users to submit their own facts and customizable preferences for fact categories.

4.2 User Authentication (Login & Signup)

The Fun Fact Generator web application includes user authentication features like login and signup, allowing users to create accounts and save their preferences for a more personalized experience. The signup page enables new users to create an account by providing necessary details, while the login page allows existing users to access their accounts securely. This ensures a secure and customized interaction with the app, enhancing the overall user experience

CHAPTER 5

CONCLUSION AND FUTURE WORK

The Fun Fact Generator Web App has been designed as an engaging platform that delivers random, interesting facts to users in a simple and user-friendly manner. By incorporating advanced filtering options and a dynamic interface, the app ensures users can personalize their experience by exploring fun facts across categories such as Science, History, and Technology. The integration of Python and Flask for backend development, coupled with MySQL for efficient database management, provides a robust infrastructure for the app's functionality.

This project demonstrates the effective use of web development technologies like HTML, CSS, and JavaScript to create an interactive and visually appealing interface. The inclusion of Python libraries like randfacts further enhances the application's capability to provide an extensive and diverse repository of facts. Overall, the web app achieves its goal of combining entertainment and education, offering users a platform to learn and enjoy simultaneously.

Future work for this project includes implementing the users to submit their own fun facts. Enable users to create profiles where they can set preferences for fact categories, favorite themes, or topics. This will make the app more personalized and engaging. Expand the filtering capabilities to include tags, popularity ranking, or trending topics. Users could also filter facts based on the length, tone (humorous, serious), or relevance to current events. Add interactive features such as quizzes, fact-based challenges, or rewards for exploring more facts. Gamification can increase user engagement and make the learning process even more exciting.

APPENDIX 1

CODING

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Fun Fact Generator</title>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@700&family=Montserrat:wght@400;700&display=swap"
rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">
    <style>
        body {
            font-family: 'Montserrat', sans-serif;
            background: linear-gradient(135deg, #f0f8ff, #e0f7fa);
            color: #333;
            text-align: center;
            margin: 0;
            padding: 50px;
        }
        nav {
            display: flex;
            justify-content: center;
            margin-bottom: 30px;
            padding: 15px;
        }
        nav a {
            margin: 0 20px;
            text-decoration: none;
            color: #4CAF50;
            font-weight: bold;
            font-size: 1.2em;
        }
        h1 {
            font-family: 'Poppins', sans-serif;
            color: #4CAF50;
            font-size: 3.5em;
            margin-bottom: 20px;
            text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);
            letter-spacing: 1px;
        }
```

```
.container {
    max-width: 800px;
    margin: auto;
    background: white;
    border-radius: 15px;
    box-shadow: 0 8px 30px rgba(0, 0, 0, 0.1);
    padding: 40px;
    transition: transform 0.3s;
}
.container:hover {
    transform: translateY(-5px);
}
.fact {
    font-size: 1.8em;
    margin: 20px 0;
    padding: 30px;
    border: 2px solid #4CAF50;
    border-radius: 15px;
    background-color: #e7ffe7;
    position: relative;
    overflow: hidden;
    transition: background-color 0.3s, transform 0.3s;
}
.fact:hover {
    background-color: #d0f0c0;
    transform: scale(1.02);
}
select, button {
    padding: 12px;
    font-size: 1.1em;
    border: 2px solid #4CAF50;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
    transition: background-color 0.3s, border-color 0.3s;
}
button {
    color: white;
    background-color: #4CAF50;
}
button:hover {
    background-color: #45a049;
    border-color: #3e8e41;
}
select:hover {
    border-color: #3e8e41;
}
footer {
    margin-top: 30px;
    font-size: 0.9em;
    color: #555;
}
```

```
        .social-icons {
            margin-top: 10px;
        }
        .social-icons a {
            margin: 0 10px;
            color: #4CAF50;
            font-size: 1.5em;
            text-decoration: none;
        }
    </style>
</head>
<body>
    <nav>
        <a href="/">Home</a>
        <a href="/login">Login</a>
        <a href="/signup">Signup</a>
    </nav>
    <div class="container">
        <h1>Fun Fact Generator</h1>
        <div>
            <label for="category">Select a category:</label>
            <select id="category">
                {% for category in categories %}
                    <option value="{{ category }}">{{ category.replace('_', ' ').capitalize() }}</option>
                {% endfor %}
            </select>
        </div>
        <div class="fact" id="fact-container">
            <strong>Did you know?</strong><br>
            <span id="fact">Select a category and click the button for a fun fact!</span>
        </div>
        <button type="button" onclick="getNewFact()">Get Another Fact</button>
    </div>

    <footer>
        <div class="social-icons">
            <a href="https://facebook.com" target="_blank"><i class="fab fa-facebook"></i></a>
            <a href="https://twitter.com" target="_blank"><i class="fab fa-twitter"></i></a>
            <a href="https://instagram.com" target="_blank"><i class="fab fa-instagram"></i></a>
            <a href="https://linkedin.com" target="_blank"><i class="fab fa-linkedin"></i></a>
        </div>
    </footer>

    <script>
        function getNewFact() {
            const category = document.getElementById('category').value;
            fetch(`/api/fact/${category}`)
                .then(response => response.json())
                .then(data => {
                    document.getElementById('fact').innerText = data.fact || "Error fetching fact.";
                })
                .catch(error => {
```

```
            document.getElementById('fact').innerText = "Error fetching fact: " + error.message;
        });
    }
  </script>
</body>
</html>
```

Login.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Attractive Login Page</title>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
      background-image: url('https://source.unsplash.com/1600x900/?forest');
      background-size: cover;
      font-family: 'Arial', sans-serif;
      color: #333;
    }

    .container {
      background: rgba(255, 255, 255, 0.9); /* More opaque for readability */
      padding: 3rem;
      border-radius: 15px; /* Softer corners */
      box-shadow: 0 10px 50px rgba(0, 0, 0, 0.3); /* More pronounced shadow */
      width: 400px;
      text-align: center;
    }

    h1 {
      color: #2E7D32; /* Green color */
      margin-bottom: 1.5rem;
      font-size: 2rem; /* Larger font size */
      font-weight: bold; /* Bolder text */
    }

    input {
      width: 100%;
      padding: 1rem; /* Increased padding */
```

```
    margin: 0.5rem 0;
    border: 1px solid #A5D6A7; /* Light green border */
    border-radius: 5px;
    font-size: 1rem;
    transition: border 0.3s, box-shadow 0.3s;
    outline: none;
}

input:focus {
    border-color: #2E7D32; /* Darker green on focus */
    box-shadow: 0 0 5px rgba(46, 125, 50, 0.5); /* Subtle glow effect */
}

button {
    width: 100%;
    padding: 1rem; /* Increased padding */
    background-color: #388E3C; /* Dark green */
    color: #fff;
    border: none;
    border-radius: 5px;
    font-size: 1.1rem; /* Slightly larger font size */
    cursor: pointer;
    transition: background-color 0.3s, transform 0.2s;
}

button:hover {
    background-color: #2E7D32; /* Slightly darker green on hover */
    transform: translateY(-2px); /* Subtle lift effect */
}

.links {
    margin-top: 1.5rem;
    font-size: 0.9rem; /* Smaller font for links */
    color: #555;
}

.links a {
    color: #388E3C; /* Dark green for links */
    text-decoration: none;
    font-weight: bold;
    transition: color 0.3s;
}

.links a:hover {
    color: #2E7D32; /* Darker green on hover */
}

/* Responsive adjustments */
@media (max-width: 480px) {
    .container {
        width: 90%; /* Full width on small screens */
        padding: 2rem; /* Reduced padding */
```

```
        }

        h1 {
            font-size: 1.6rem; /* Smaller heading */
        }
    }
  </style>
</head>
<body>

  <div class="container">
    <h1>Login to Your Account</h1>
    <form>
      <input type="email" placeholder="Email Address" required>
      <input type="password" placeholder="Password" required>
      <button type="submit">Login</button>
      <div class="links">
        <span>Don't have an account?</span>
        <a href="signup.html"> Sign Up</a>
      </div>
    </form>
  </div>

</body>
</html>
```

Signup.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Attractive Signup Page</title>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
      background-image: url('https://source.unsplash.com/1600x900/?forest');
      background-size: cover;
      font-family: 'Arial', sans-serif;
      color: #333;
    }
```

```
.container {
    background: rgba(255, 255, 255, 0.9); /* More opaque for readability */
    padding: 3rem;
    border-radius: 15px; /* Softer corners */
    box-shadow: 0 10px 50px rgba(0, 0, 0, 0.3); /* More pronounced shadow */
    width: 400px;
    text-align: center;
}

h1 {
    color: #2E7D32; /* Green color */
    margin-bottom: 1.5rem;
    font-size: 2rem; /* Larger font size */
    font-weight: bold; /* Bolder text */
}

input {
    width: 100%;
    padding: 1rem; /* Increased padding */
    margin: 0.5rem 0;
    border: 1px solid #A5D6A7; /* Light green border */
    border-radius: 5px;
    font-size: 1rem;
    transition: border 0.3s, box-shadow 0.3s;
    outline: none;
}

input:focus {
    border-color: #2E7D32; /* Darker green on focus */
    box-shadow: 0 0 5px rgba(46, 125, 50, 0.5); /* Subtle glow effect */
}

button {
    width: 100%;
    padding: 1rem; /* Increased padding */
    background-color: #388E3C; /* Dark green */
    color: #fff;
    border: none;
    border-radius: 5px;
    font-size: 1.1rem; /* Slightly larger font size */
    cursor: pointer;
    transition: background-color 0.3s, transform 0.2s;
}

button:hover {
    background-color: #2E7D32; /* Slightly darker green on hover */
    transform: translateY(-2px); /* Subtle lift effect */
```

```
        }

        .links {
            margin-top: 1.5rem;
            font-size: 0.9rem; /* Smaller font for links */
            color: #555;
        }

        .links a {
            color: #388E3C; /* Dark green for links */
            text-decoration: none;
            font-weight: bold;
            transition: color 0.3s;
        }

        .links a:hover {
            color: #2E7D32; /* Darker green on hover */
        }

        /* Responsive adjustments */
        @media (max-width: 480px) {
            .container {
                width: 90%; /* Full width on small screens */
                padding: 2rem; /* Reduced padding */
            }

            h1 {
                font-size: 1.6rem; /* Smaller heading */
            }
        }
    </style>
</head>
<body>

    <div class="container">
        <h1>Create Account</h1>
        <form>
            <input type="text" placeholder="Full Name" required>
            <input type="email" placeholder="Email Address" required>
            <input type="password" placeholder="Password" required>
            <input type="password" placeholder="Confirm Password" required>
            <button type="submit">Sign Up</button>
            <div class="links">
                <span>Already have an account?</span>
                <a href="login.html"> Login</a>
            </div>
        </form>
```

```
    </div>

</body>
</html>
```

App.py:

```
from flask import Flask, render_template, jsonify, request, redirect, url_for, flash
import random

app = Flask(__name__)
app.secret_key = 'your_secret_key'  # Required for session management and flash messages

# Fun facts categorized
fun_facts = {
    "funny": [
        "In Switzerland, it’s illegal to own just one guinea pig.",
        "A shrimp's heart is located in its head.",
        "Cows have best friends and get stressed when they are separated.",
        "Apples float because they are 25% air.",
        "A shrimp's heart is located in its head."
    ],
    "animal_facts": [
        "Axolotls can regenerate their limbs.",
        "Octopuses have blue blood and three hearts.",
        "The heart of a blue whale is so big that a human can swim through its arteries.",
        "It is physically impossible for pigs to look up into the sky.",
        "An ostrich's eye is bigger than its brain.",
        "Ants take rest for around 8 minutes in a 12-hour period.",
        "Humans are the only animals that blush.",
        "Dolphins have names for each other and can call each other by name.",
        "Starfish have no brain."
    ],
    "pop_culture": [
        "The first film ever to be made was 'Roundhay Garden Scene' in 1888.",
        "The longest-running TV show is 'The Simpsons'.",
        "The first iPhone was released in 2007.",
        "Mickey Mouse was originally named Morty."
    ],
    "food": [
        "Honey never spoils; archaeologists have found pots of honey in ancient Egyptian tombs that are over 3000 years old and still edible.",
        "Ketchup was sold in the 1830s as medicine.",
        "Chocolate was once used as currency by the Aztecs."
    ],
    "science": [
        "A day on Venus is longer than a year on Venus.",
```

```
        "One million Earths could fit inside the sun.",
        "The strongest muscle in the body is the tongue."
    ],
    "historical_facts": [
        "The Mona Lisa has no eyebrows.",
        "Competitive art used to be an Olympic sport.",
        "The Eiffel Tower can be 15 cm taller during the summer."
    ]
}

@app.route('/')
def index():
    return render_template('index.html', categories=fun_facts.keys())

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        # Here you would typically validate the username and password
        flash('Login successful!')  # Replace with actual login logic
        return redirect(url_for('index'))
    return render_template('login.html')

@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        # Here you would typically save the new user to a database
        flash('Signup successful! Please log in.')  # Replace with actual signup logic
        return redirect(url_for('login'))
    return render_template('signup.html')

@app.route('/api/fact/<category>', methods=['GET'])
def get_fact(category):
    category = category.lower().replace(' ', '_')
    if category in fun_facts:
        fact = random.choice(fun_facts[category])
        return jsonify(fact=fact)
    else:
        return jsonify(error="Category not found"), 404

if __name__ == '__main__':
    app.run(debug=True)
```

APPENDIX 2

SNAPSHOTS

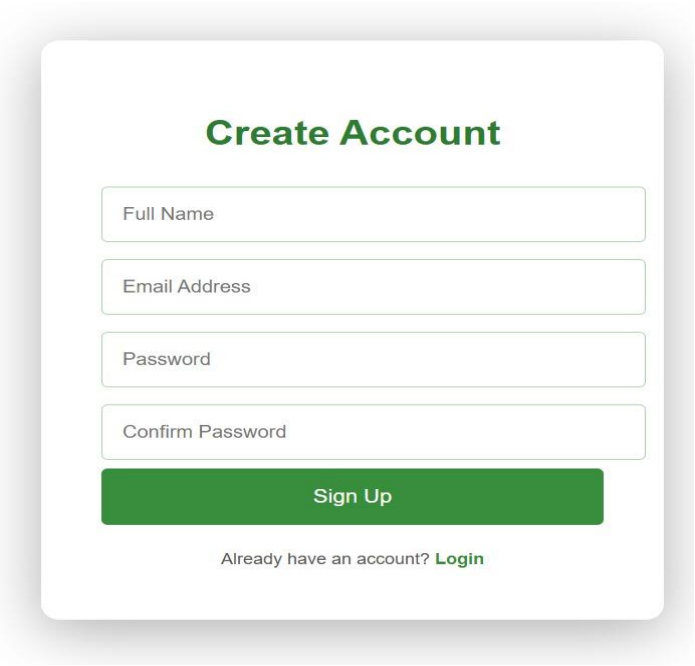

Figure A2.1 SIGNUP PAGE

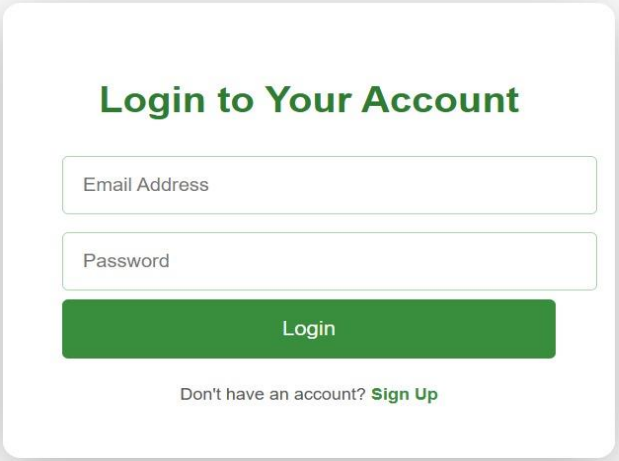

Figure A2.2 LOGIN PAGE

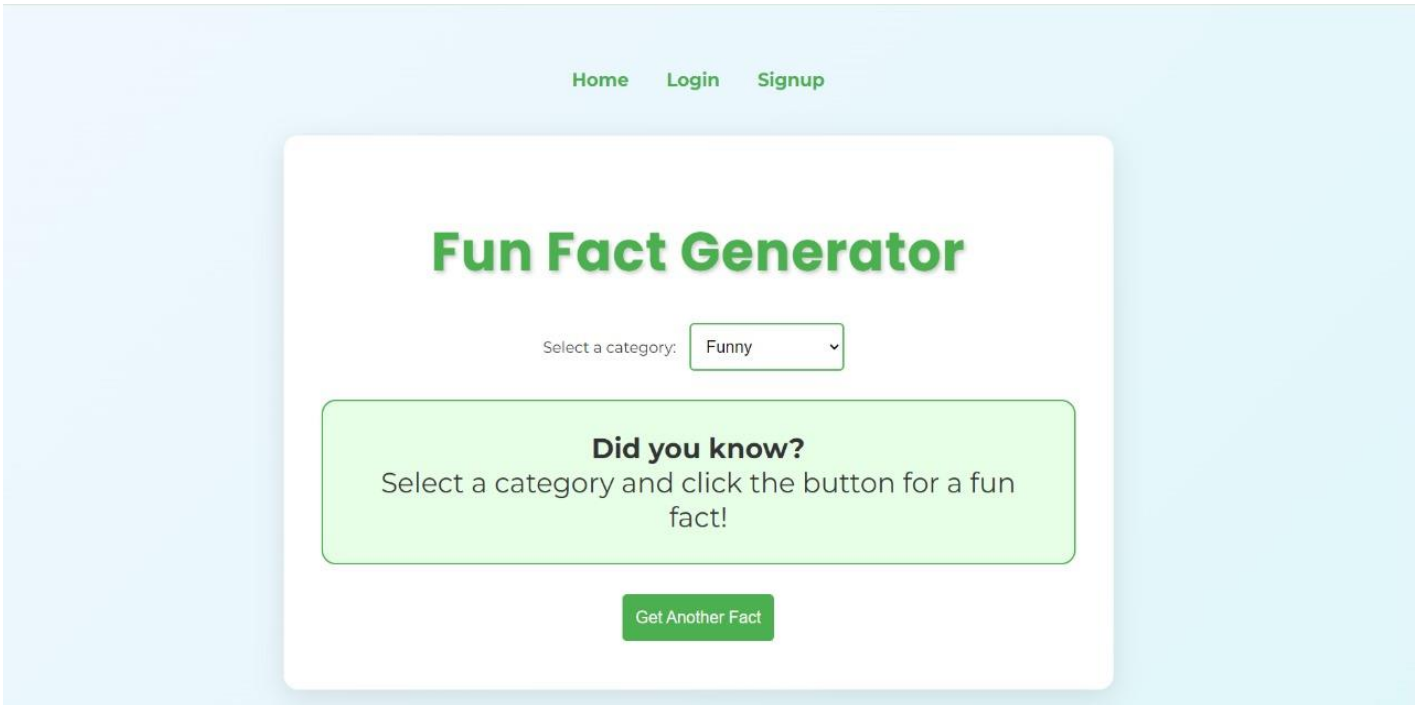

Figure A2.3 FUNFACT GENERATOR WEB APP

Figure A2.4 FUNFACT GENERATOR WEB APP

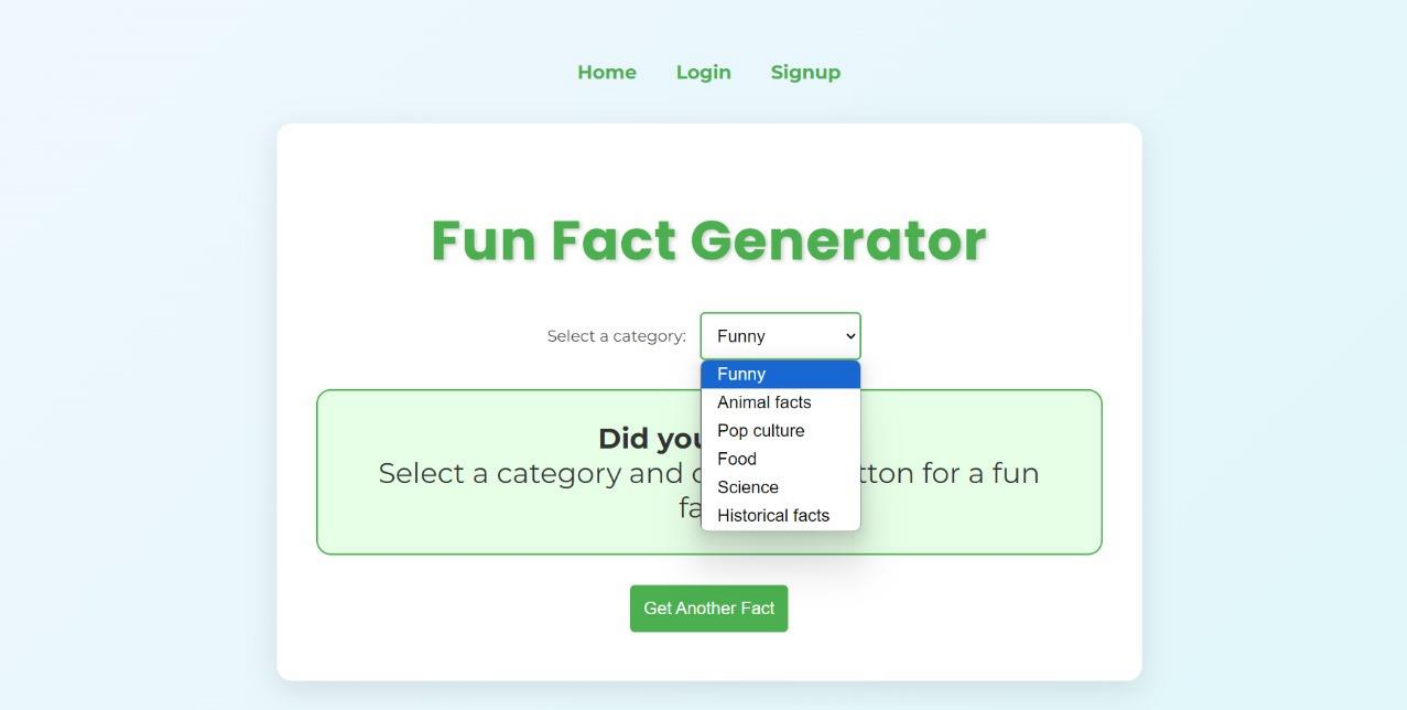

Figure A2.5 FUNFACT GENERATOR WEB APP

REFERENCES

[1]. https://www.geeksforgeeks.org/fun-fact-generator-web-app-in-python/

[2].https://github.com/someuser/fun-fact-generator

[3].https://www.tutorialspoint.com/fun-fact-generator-web-app-in-python

[4].https://pypi.org/project/randfacts/

[5]. https://www.javatpoint.com/flask-tutorial

[6].https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3

[7].https://lyz-code.github.io/blue-book/python_jinja2/

[8].https://medium.com/bhavaniravi/build-your-1st-python-web-app-with-flask-b039d11f101c

[9].https://www.w3schools.com/htmL/html_scripts.asp

[10].https://jinja.palletsprojects.com/en/stable/