EXPLORE 360 REPORT TEMPLATE

1.Problem Definition and Backstory

A. Why?

Wolfville already has an official town website that shares basic information about places to visit, events, and services (wolfville.ca). However, during our research, we found that the site is more formal and not very engaging for students or visitors who want to explore the town in a fun and simple way.Our group decided to create **Explore 360**, a local guide website made specifically for Wolfville. Our goal was to create a local guided website for Wolfville, providing a comprehensive directory of places to visit, restaurants, events, and activities in town.The purpose of Explore 360 is to act as a central hub where residents, tourists, and students can easily discover what Wolfville has to offer. It focuses on real-life experiences and is designed to be easy to use.

What Makes Explore 360 Different:

- A complete list of local businesses (restaurants, hotels, shopping, things-to-do, events, guidelines, history, reviews, contacts) with custom data.
- User-submitted reviews and star ratings to help others decide where to go.
- A working review system that supports Create, Read, Update, and Delete actions.
- A search feature that lets users find establishments quickly.
- A design that feels modern and is more relatable for students and visitors.

We chose not to use Nova Scotia's Open Data API because the data was outdated and didn't fit our needs. Instead, we created our own database using **MySQL**, which gave us more control and allowed us to add just the right amount of entries (37 establishments total).

b. Learning Objectives

Front-End Learning:
We learned how to use HTML for building the structure of our web pages. We used semantic tags to help with navigation,and keep our code clean and easy to understand.With CSS, we styled the layout of the website including the buttons, search bars, images, and page responsiveness. We focused on a layout that works on different screen sizes.We used external stylesheets to separate

design from content, and followed CRAP principles (Contrast, Repetition, Alignment, Proximity) to make the site look better.

Back-End Learning:
On the back-end, we used PHP to connect our website to the MySQL database and handle form inputs.We implemented CRUD operations for the review system. Users can submit a review through a form, and the review gets stored and displayed**.**We learned how to read data from the database and show it on the website in real-time using files like **showReviews.php** and **showEstablishments.php.**

Teamwork Project Experience:
We practiced working in a team where each person had their own tasks and we had to merge everything into one website.We communicated regularly and shared code to make sure both front-end and back-end worked smoothly together.

2. Success Criteria (What was the MVP, Why?)

For Explore 360 (Wolfville Guide), our Minimum Viable Product (MVP) focused on essential features that would help new students and visitors navigate Wolfville more easily.

As international students, we faced challenges in finding interesting places, restaurants that suited our palettes, and general navigation around town. Without friends or family to guide us, discovering Wolfville's attractions was difficult.

To solve this, our MVP included:

- **Home Page:** Provides an overview of key attractions in Wolfville.
- **List of Places:** Categorized locations (restaurants, parks, events, etc.).
- **Basic Search & Filter:** Helps users quickly find places based on type.
- **Location Details Page:** Includes descriptions and images for better understanding.
- **Responsive Design:** Ensures accessibility on both mobile and desktop.

We chose these core features to offer immediate value while keeping development manageable. They lay the foundation for expanding the platform with more interactive and personalized features in the future.

3. Full Stack Technologies and Architecture Used and Why:

Front-End Technologies:

The front end was developed using **HTML** and **CSS**. As being one of the most fundamental languages for structuring web content, HTML was a must have. Also, though Bootstrap was a valid contender for front-end design, CSS proved itself to be more flexible and customizable. Styling with CSS was more time consuming, and had a significant learning curve, of course, but it did allow us to achieve the precise look we aimed for. CSS excels at targeting specific design requirements, and handling minute details, which would have been more restrictive if we opted for Bootstrap instead.

Back-End Technologies:

The Back-End was built with **PHP** and **MySQL**, with **Apache** as our web server. We used the **AMPPS** stack to handle our configuration and act as our development environment. Several of our team members with Windows operating systems experienced some difficulty with the setup at first, but with a simple alteration in the php configuration file, we were able to resolve the issue. The AMPPS environment was user friendly and well suited to manage our database relations, and PHP worked seamlessly and powerfully with our other back-end components. In addition, it was useful to have the PHP documentation at our fingertips with the all inclusive installation of AMPPS.

Moreover, we implemented **jQuery** and **AJAX** (JavaScript) to provide our website with a seamless and dynamic display of user reviews as they were updated. One of our primary features was the creation and viewing of reviews for each establishment. It was imperative that our system would be able to automatically update and display user changes in real-time. AJAX was perfect for this and worked hand in hand with our PHP layout.

Database Technologies:

We considered using a stack like MERN or MEVN, with MongoDB as our database, but found that **MySQL** better suited our needs for a few reasons. Firstly, MySQL was already included within the environment (AMPPS) that we used and were familiar with. Secondly, our database would need to be uniform where all entries would share the same attributes (eg. all establishments had a name, address, rating etc). Thus, we preferred a relational database. Moreover, MySQL worked well for us due to its referential capacity. Our system involves two database tables: establishment and review. We needed to link each establishment with their respective reviews, and so the foreign key functionality within MySQL was a simple, yet robust solution to our needs.

4a. Group Member Contributions

The development of the full-stack project was a collaborative effort, with each team member contributing to different aspects of the execution. The front-end and back-end were split between 3 members of the group respectively. The project was divided into various sections, ensuring a structured and efficient workflow. Below is a breakdown of the contributions:

- **Divine-Favour Damasus** – Divine focused on the **front-end development** and **UI/UX design** of the project. She ensured that the interface was visually appealing, intuitive, and accessible.
- **Kaylee Smith** – Kaylee worked on **back-end development**, responsible for organizing data gathering, setting up the database, and handling user reviews. She ensured that the database tables were secure and accurate.
- **Melina Bizimana Uwase** – Melina worked on **back-end development**, contributing significantly to the **search and filter functionality**, ensuring users could easily navigate and find relevant information. She also worked on dynamic content rendering to provide a seamless user experience.
- **Jannatul Baki** – Jannatul worked on the **front-end development,** compiled a structured breakdown specifying a list of webpages, what each webpage should cover and where we can pull the data from.
- **Shekin David Raj** – Shekin worked on the **front-end development,** She worked on security enhancements to protect user data.
- **Rosemary Etu-Efeotor** – Rosemary played a crucial role in **full-stack integration**, ensuring the frontend and backend components worked seamlessly. She also handled the **debugging and testing process**, identifying issues and ensuring functionality before deployment.

b. Debugging and Testing

Testing and debugging were essential to ensuring the project's stability and reliability. The team employed various strategies to identify and resolve issues:

Frontend Testing:

- Conducted **cross-browser testing** to ensure the application functioned well across Chrome, Firefox, Edge, and Safari.
- Used **mobile responsiveness testing** tools to check layout adaptability on different devices.
- Verified user interface consistency by adjusting CSS properties and HTML configurations.

Backend Testing:

- Implemented **unit tests** on core backend functionalities like reading and creating user reviews, database queries, and data validation.

Database Testing:

- Ensured **data integrity and consistency** by testing CRUD (Create, Read, Update, Delete) operations with user reviews.
- Optimized SQL queries to improve database response time.
- Used indexing to improve the performance of search functionalities.

Unit Testing:

- Conducted tests on individual components and modules to catch bugs before integration.
- Ensured proper error handling in the backend by testing how the system responded to invalid inputs.

User Testing:

- Collected feedback on navigation, ease of use, and performance.
- Made iterative improvements based on user suggestions.

c. Difficulties and Solutions

Throughout the development process, the team encountered several challenges. Below are the major difficulties faced and the solutions implemented:

Google Maps API Integration Issues

I. **Problem:** Implementing Google Maps API presented a challenge due to API key restrictions and billing limitations. Some functionalities required a paid version, which was not within our initial budget.

Solution: When we realized this problem, we opted to exclude Google Maps API from our project.

II. Time Constraint

Problem: Due to time constraints, we were unable to implement the admin functionality in our project. While it was part of our initial plan, other core features took priority, and we did not have sufficient time to complete the admin section.

5.)**Final Implementation Report**

1. Main Features

a) User Interface (Front-End)

· **Home Page:** Provides an overview of available categories such as hotels, restaurants, shopping, events, and things to do.

· **Navigation Sidebar:** A collapsible sidebar for easy access to different sections.

· **Search Functionality:** Integrated search bar for quick lookup of establishments.

· **Visual Enhancements:**

o Use of CSS for structured layouts and improved aesthetics.

o Responsive design ensuring usability across different devices.

o Image display and alignment for better user experience.

· **Review System:**

o Users can submit reviews.

o Star rating system for user feedback.

b) Back-End Functionality

· **Database Structure:**

o The `establishments.sql` file defines a MySQL database (`establishments_db`) containing:

§ `establishment` table (stores details like name, address, rating, and category).

§ `review` table (links reviews to establishments via `establishment_name` as a foreign key).

· **Review Management System:**

o Users can submit reviews, which are stored in a MySQL database (`reviews.sql`).

o Reviews can be retrieved and displayed dynamically on the frontend.

· **PHP Scripts:**

o `showEstablishments.php` fetches establishment details from the database.

o Implements a search feature (`$_GET['search']`).

o Connects to `establishments_db` using a MySQL connection.

o Retrieves establishment names, ratings, addresses, and categories.

- **API Endpoints for Review Display:**

o The `showReviews.php` script fetches reviews from the database and presents them to the frontend.

- **Secure Data Handling:**

o PHP scripts interact with the database using SQL queries.

o Basic input handling and retrieval mechanisms are implemented.

2.)Adherence to Your Group's Proposed MVP

Concept Overview:
Explore360 is a local guide website for Wolfville that provides a comprehensive directory of places to visit, restaurants, events, and activities. It is intended for residents, students, and tourists to explore Wolfville's offerings. The platform organizes content into categories and features user-friendly navigation, search, and potential for future scalability.

MVP Requirements and Implementation:

MVP Feature	**Status**	**Implementation Detail**
Home Page: Overview of Wolfville attractions	✔	Implemented with a welcoming layout introducing the platform.
List of Places: Restaurants, parks, events, etc.	✔	Implemented through HTML pages for hotels, restaurants, shopping, events, and more.
Basic Search & Filter	✔	Search functionality integrated in showEstablishments.php. Event filtering included.

Location Details Page	✔	Each establishment includes name, rating, address, and description.
Responsive Design	✔	CSS ensures compatibility with mobile and desktop.

Nice-to-Have Features:

Feature	**Status**	**Notes**
User Reviews & Ratings	✔	Implemented with reviews.php and showReviews.php.
Event Listings	✔	Events page with images, descriptions, and filtering.
Dark Mode UI	✘	Not yet implemented.

Stretch Goals:

Feature	**Status**	**Notes**
User Accounts & Favourites	✘	Not implemented.
AI-Powered Recommendations	✘	Not implemented.
Google Maps API Integration	✘	Not implemented.

6)Conclusions
Final Results vs. Proposal and Future Work

Overview
This website aims to guide tourists exploring Wolfville by providing extensive information and easy access to essential services.

DETAILS:
Page 1: Home
Provides a brief introduction and description of Wolfville.

Page 2: Hotels
Displays a list of hotel options available for tourists.

Each item in the list includes an image, name, and address.

Page 3: Restaurants
Lists a variety of restaurants and cafes in Wolfville.
Each item includes a picture, description of the cuisine, address, opening and closing hours.

Page 4: Shopping
Lists all stores and shops in town, including descriptions, addresses, and business hours.

Page 5: Things to Do
Offers a list of tourist attractions, places of interest, and activities in Wolfville.
Each item includes images, a description, reviews and ratings, and the address.
Clicking on an item gives a gallery of additional images.

Page 6: Events
Provides a list of upcoming events in Wolfville, including event names, dates, and locations.

Page 7: Guidelines
Offers helpful guidance on traveling to Wolfville, including transportation options, taxi services, ideal visiting times, and emergency procedures.

Page 8: History
Provides a brief history of Wolfville, emphasizing its cultural and historical significance.

Page 9: Reviews
Shows reviews from tourists who have visited Wolfville.

Page 10: Contacts
Lists emergency contacts, essential phone numbers, and addresses for urgent situations.

FINAL RESULT

1.The Home Page has been successfully developed, meeting all requirements. Additionally, a search bar has been added at the top of the page for enhanced navigation.

2.On the Hotels Page, we have provided a list of hotels in Wolfville, each accompanied by an image and a brief description. To meet the deadline, Airbnb options were excluded, and links to hotel websites were not connected in this version.

3.The Restaurants Page features a list of cafes and restaurants in Wolfville, with each item displaying a picture, description, and essential details. For now, direct links to the restaurants' official websites have not been included.

4.On the Shopping Page, users can browse a list of local stores, each with a brief description. Links to the stores' websites were not added in this version.

5.The Things to Do Page has been successfully implemented, listing out a variety of attractions, activities, and places of interest in Wolfville.

6.The Reviews Page includes a feature to leave reviews for a specific establishment found on the site (via a dropdown list), as well as the display of past reviews from previous users. Users do not require authentication, but may leave anonymous reviews.

7.The Events Page shows a detailed list of upcoming events in Wolfville, including event names, dates, and locations as planned.

8.The Map Page is not developed in this version of the website.

9.The Guideline Page was successfully implemented according to the proposal. It provides useful advice on traveling to Wolfville, where to find taxis, the best times to visit, and steps to take in emergencies.

10.The History Page shows Wolfville's rich history and was implemented as planned.

11.The Contacts Page lists emergency contacts, phone numbers, and addresses for use in urgent situations. This page was also successfully completed as per the proposal.

FUTURE WORK:-
For the next updates of the website we are going to:

1. Add Direct links to the official websites of hotels, restaurants, and shops. Implementing these links would allow users to make bookings, place orders, and explore stores easily.
2. Will add the maps page to provide live location and directions.
3. Add Airbnb's to the Hotels page which would provide more accommodation choices.
4. The Things to Do Page will include user reviews and ratings for attractions and activities.
5. The Reviews Page will be able to collect and display feedback from visitors.
6. Options for user accounts, sign-up, and login will be available. This is essential for enabling reviews, ratings, and personalized experiences.

7. References and Resources Used
➔Used W3schools for support in coding.
➔Adobe Illustrator for initial design format.
➔Current town of Wolfville website for reference and events https://wolfville.ca/
➔Used all the stores / hotels / shops in town to make list and get their locations.
➔Various YouTube Support Videos:

- https://www.youtube.com/watch?v=ejN-oAw9vC0
- https://www.youtube.com/watch?v=Y9yE98etanU&t=980s
- https://www.youtube.com/watch?v=irP-aSgGXZQ