

# Simple Survey

## Overview

Survey built with React, Vite, Node.js, JavaScript, HTML, CSS, SQLite which has a user enter their name and answer a series of questions to determine whether they are a vacationer or a traveler and storing their name and score in a database.

---

## High-Level Overview

When a user opens the website, they are greeted with a welcome screen. After clicking next, they are taken to a page where they can enter their name. Once they enter their name and go to the next page, they answer a series of questions which help determine whether they are a vacationer or a traveler. If they select an option which matches a vacationer, their score doesn't increase. However, if they select an option which matches a traveler, their score increases by 1. After submitting their answers, on the final page, the user sees their name, their score and their category. If they achieved a score of 0-6, they are a vacationer. If they achieved a score of 7-10, they are a traveler. The user's name and score are stored in a database.

---

## Motivation

This project was completed as a personal project to gain hands-on experience with fullstack development in React and Node.js.

---

## Technologies & Tools

- **Programming Languages:** JavaScript, HTML, CSS
- **Frameworks/Libraries:** React, Node.js
- **Tools:** VSCode, Git, Github, Vite
- **Databases:** SQLite

---

## System Design

4 Main Screens:

- Initial Welcome Screen
    - Welcomes the user and allows them to click next to start entering their name
  - Name Input Screen
    - Allows the user to enter in their name
  - Survey Screen
    - A list of questions containing 2 answer choices: one which classifies you as a vacationer and another as a traveler
  - Results Screen
    - Displays your name, score and classification based on the questions answered previously
- 

## Key Features

- Data stored in binary format within a SQLite database
  - User data entry and storage capabilities
  - Error handling
  - Score calculation capabilities
- 

## Challenges & Solutions

Challenge	How I Addressed It
Redundant question handling logic	Stored questions in an array and had logic repeated per question
Offline data access	Stored data in a local database

---

## Results

Successfully created a fully-functional survey which asks users questions and stores their data in a SQLite database in binary format