This week's assignment involves creating a JavaBeans project to hold data, utilizing a JSP page to gather and display this data in a well-structured format. The JavaBean must implement the java.io.Serializable interface, and the JSP page should include scriptlets for Java code, with all HTML tags placed outside of these scriptlets. The data input should consist of at least five fields and the output should be formatted in an HTML table, including additional descriptive elements.

To complete this assignment, we will follow a structured approach that includes creating a JavaBean, setting up a JSP page, and ensuring that the data is displayed correctly.

Step 1: Create the JavaBean

A JavaBean is a reusable software component that follows specific conventions. It must have a public no-argument constructor and properties that can be accessed via getter and setter methods, and it should implement the Serializable interface.

*TravelDestination.java*

A screenshot of a computer program

AI-generated content may be incorrect.

Code Structure

The structure of the TravelDestination.java class is straightforward and follows Java conventions. Here is a breakdown of its components:

* **Package Declaration**: The class is part of the com. example.travel package, which helps in organizing related classes.
* **Imports**: The Serializable interface is imported to allow the class to be serialized.
* **Class Definition**: The class is defined with the public access modifier, making it accessible from other classes.
* **Fields**: Three private fields are declared to hold the destination's country, city, and description.
* **Constructors**: A no-argument constructor is provided, allowing for the creation of instances without requiring the setting of initial values.
* **Getters and Setters**: Public methods are defined to retrieve and update the values of private fields.

Step 2: Create a List of JavaBeans

index.jsp

**A white page with black text

AI-generated content may be incorrect.**

Code Structure

index.jsp is structured as follows:

* **Directives**: The code begins with JSP directives that define the content type and import necessary Java classes.
* **HTML Structure**: The HTML document is defined by a head and body section, which includes metadata and links to external stylesheets.
* **Java Code**: Embedded Java code initializes personal information and a list of travel destinations, which are then displayed in an HTML table.

Step 3 The styles.css to polish our data display

A screenshot of a computer program

AI-generated content may be incorrect.

Code Structure

styles.css is structured in a way that each section targets specific HTML elements.

**Global Styles**: The body styles set the overall look and feel of the page.

* **Heading Styles**: The h1 styles specifically target the main headings.
* **Table Styles**: The table, th, and td styles focus on the presentation of tabular data.

This modular approach allows for easy maintenance and updates to the stylesheet.

The separation of Java code and HTML ensures clarity and maintainability, while the use of a JavaBean allows for a structured approach to data management.