**1- A project description (Introduction, scope, similar products, requirments, features design etc)**

The Travel Agency Management System, named "Epoka Travel," is a comprehensive web-based platform designed to streamline the operations of a travel agency. The system facilitates package management, customer bookings, and administrative oversight, providing an efficient solution for both customers and administrators. The platform features a user-friendly interface that allows customers to browse travel packages, manage their shopping carts, and create accounts, while administrators can manage content, view orders, and monitor payments.

The scope of this project encompasses the development of a full-fledged travel agency website with functionalities for customers and administrators. Customers can explore various travel packages, view details such as costs and destinations, add items to their shopping carts, and proceed to checkout. The system also supports user registration, where customers can create accounts by providing personal and contact information. Administrators have access to a dedicated panel where they can insert and manage packages, categories, types, employees, and customers, as well as monitor orders and payments.

Similar products in the market include platforms like Expedia, Booking.com, and local travel agency websites. These systems offer package browsing, booking, and payment processing. However, our app distinguishes itself by focusing on a simplified user experience and a robust administrative backend tailored for small to medium-sized travel agencies. Unlike global platforms, this system is designed to cater to specific regional offerings, such as packages for Darjeeling, Taj Mahal, and Sri Lanka, as highlighted in the screenshots.

The system requires a server-side scripting language (e.g., PHP), a database (e.g., MySQL), and a web server (e.g., Apache) to function. Key functional requirements include:

* *Customer Interface:* Browse packages, view details, add to cart, and checkout.
* *User Registration:* Capture customer details such as name, email, passport ID, and address.
* *Admin Panel:* Manage packages, categories, types, employees, and orders.  
  Non-functional requirements include responsive design, secure payment processing, and data privacy compliance.

The system offers the following features:

1. Package Browsing: Customers can view travel packages categorized by business, economy, and regular types.
2. *Shopping Cart:* Users can add packages to their cart, update quantities, and remove items.
3. *User Accounts:* Customers can register by providing personal and travel-related details.
4. *Admin Dashboard:* Administrators can insert, update, and delete packages, categories, and employee records.
5. *Order Management:* Admins can view and track customer orders and payments.

The design follows a modular structure with separate interfaces for customers and administrators. The customer interface includes a homepage, package listings, a shopping cart, and a registration form. The admin panel provides navigation options to manage various aspects of the system, such as packages, categories, and employees. The UI is straightforward, with clear headings and intuitive navigation menus, as seen in the screenshots.

The platform prioritizes ease of use, with a clean layout and logical flow. Customers can quickly locate packages, view costs (e.g., $5000 for Darjeeling), and proceed to book. The shopping cart displays the total items and price prominently, enhancing transparency. The admin panel is organized into sections, ensuring efficient content management.

The system is likely built using PHP for server-side logic, MySQL for database management, and HTML/CSS for the front end. The cart.php page (shown in screenshot 2) handles dynamic updates to the shopping cart, while the admin panel (screenshot 4) relies on role-based access control to restrict functionalities to authorized personnel.

Potential challenges include ensuring secure payment processing, managing high traffic during peak booking seasons, and maintaining data accuracy across packages and customer records. The system must also handle concurrent user sessions without performance degradation. Future updates could integrate real-time package availability checks, customer reviews, and a loyalty program. Additionally, incorporating AI-based recommendations for personalized travel suggestions could enhance user engagement. The system may also expand to include multi-language support and mobile app compatibility for broader accessibility.

*Functional Requirements (What the system should do)*

1. *User Registration & Authentication*
   * Customers must be able to create an account by providing personal details (name, email, password, passport ID, address, etc.).
   * Users should be able to log in and log out securely.
2. *Package Browsing & Search*
   * Customers should be able to view available travel packages (e.g., Sri Lanka, Darjeeling, Taj Mahal).
   * The system must support filtering by categories (Business, Economy, Regular) and types (Single, Family, Couple).
3. *Shopping Cart Management*
   * Users should be able to add, remove, and update the quantity of packages in their cart.
   * The cart must display the total items and price in real-time.
4. *Checkout & Booking*
   * Customers should be able to proceed to checkout, confirm bookings, and receive a summary of their order.
   * The system should generate an order confirmation (e.g., via email or on-screen).
5. *Admin Panel for Content Management*
   * Administrators must be able to add, edit, and delete travel packages, categories, and types.
   * Admins should manage customers, employees, orders, and payments from a centralized dashboard.
6. *Order & Payment Tracking*
   * Admins should view all customer orders, payment statuses, and booking details.
   * Customers should see their order history after logging in.
7. *Responsive & Accessible UI*
   * The website should be usable on desktop and mobile devices with a consistent layout.
   * Critical actions (e.g., checkout, login) should be easily accessible.
8. *Search Functionality*
   * Users should be able to search for packages by destination, price range, or keywords.
9. *User Profile Management*
   * Customers should update their personal information, view booking history, and manage preferences.
10. *Security & Data Protection*

* User passwords must be encrypted (e.g., hashed).
* Payment information (if integrated) should comply with PCI-DSS standards.

*Non-Functional Requirements (Quality & Performance Standards)*

1. *Performance*
   * The system should load pages within **2-3 seconds** under normal traffic.
   * The shopping cart must update quantities and totals without delays.
2. *Scalability*
   * The database should handle **100+ concurrent users** without crashing.
   * The system should support future expansion (e.g., adding new destinations).
3. *Security*
   * SQL injection and XSS attacks must be prevented.
   * Admin access should require strong authentication (e.g., role-based permissions).
4. *Usability*
   * The interface should follow intuitive navigation (e.g., clear menus, buttons).
   * Error messages should guide users (e.g., "Invalid email format").
5. *Reliability*
   * The system should have **99% uptime** (excluding maintenance).
   * Backup and recovery mechanisms must be in place.
6. *Compatibility*
   * The website should work on **Chrome, Firefox, Safari, and Edge**.
   * Should adapt to different screen sizes (mobile, tablet, desktop).
7. *Data Integrity*
   * No duplicate bookings for the same package and user.
   * Cart data should persist for logged-in users across sessions.
8. *Accessibility*
   * The website should support screen readers and keyboard navigation.
   * Text contrast should meet **WCAG 2.1 standards**.
9. *Maintainability*
   * The codebase should be well-documented for future updates.
   * Database schema should allow easy modifications.
10. *Regulatory Compliance*

* If handling payments, the system must comply with **GDPR** (for EU users) and local data protection laws.

**2- Database Schema (Table names, Columns, Data types, Keys)**

The database tagency consists of 7 primary tables that manage users, packages, categories, types, customers, employees, and shopping cart data. The schema is designed to support a travel agency system with functionalities for package browsing, booking, user management, and administrative control.

**Table:**admins**(Administrator Accounts)**

* **Purpose:** Stores login credentials for system administrators.
* **Columns:**
  + user\_id (INT, PRIMARY KEY): Unique admin identifier.
  + user\_email (VARCHAR): Admin's email (e.g., admin@gmail.com).
  + user\_pass (VARCHAR): Hashed password (e.g., admin).

**Table:**cart**(Shopping Cart)**

* **Purpose:** Tracks items added to user carts.
* **Columns:**
  + p\_id (INT, PRIMARY KEY): Package ID linked to packages.package\_id.
  + ip\_add (VARCHAR): User’s IP address.
  + qty (INT): Quantity of the package (default: 1).
* **Notes:** Uses IP addresses for guest users; logged-in users likely reference customers.customer\_id.

**Table:**categories**(Package Categories)**

* **Purpose:** Defines package categories (Business, Economy, Regular).
* **Columns:**
  + cat\_id (INT, PRIMARY KEY): Category ID.
  + cat\_title (TEXT): Category name (e.g., Business).

**Table:**customers**(User Accounts)**

* **Purpose:** Stores customer registration details.
* **Columns:**
  + customer\_id (INT, PRIMARY KEY): Unique user ID.
  + customer\_ip (VARCHAR): IP address.
  + customer\_name (TEXT): Full name.
  + customer\_email (VARCHAR): User’s email.
  + customer\_pass (VARCHAR): Hashed password.
  + c\_passport (VARCHAR): Passport ID.
  + customer\_country (TEXT), customer\_city (TEXT): Location details.
  + customer\_contact (VARCHAR): Phone number.
  + customer\_address (TEXT): Physical address.
  + customer\_image (TEXT): Profile image path.

**Table:**employees**(Staff Records)**

* **Purpose:** Manages travel agency employees (e.g., local agents).
* **Columns:**
  + emp\_id (INT, PRIMARY KEY): Employee ID.
  + emp\_name (VARCHAR), emp\_email (VARCHAR): Personal details.
  + emp\_designation (VARCHAR): Role.
  + emp\_location (VARCHAR), emp\_address (TEXT): Work location.
  + emp\_contact (VARCHAR): Contact number.

**Table:**packages**(Travel Packages)**

* **Purpose:** Stores package details (destinations, prices, descriptions).
* **Columns:**
  + package\_id (INT, PRIMARY KEY): Unique package ID.
  + package\_cat (INT): Foreign key to categories.cat\_id.
  + package\_type (INT): Foreign key to types.type\_id.
  + package\_title (VARCHAR): Name.
  + package\_price (INT): Cost in USD.
  + package\_desc (LONGTEXT): HTML-formatted description.
  + package\_image (TEXT): Image path.
  + package\_keywords (VARCHAR): Search keywords.
* **Relationships:**
  + Linked to categories and types tables for filtering.

**Table:**types**(Package Types)**

* **Purpose:** Defines package types (Single, Family, Couple).
* **Columns:**
  + type\_id (INT, PRIMARY KEY): Type ID.
  + type\_title (TEXT): Type name (e.g., Family).

**Key Observations**

1. **Security:**
   * Passwords are stored in plain text (e.g., admin), which violates security best practices. **Recommendation:** Use hashing (e.g., bcrypt).
2. **Data Relationships:**
   * packages.package\_cat → categories.cat\_id.
   * packages.package\_type → types.type\_id.
   * cart.p\_id → packages.package\_id.
3. **Missing Foreign Keys:**
   * The schema lacks explicit FOREIGN KEY constraints, relying on application logic.

**3- Some print screens from the website (frontend and backend)**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a website

AI-generated content may be incorrect.

**4- Some parts of the source code**

<?php

session\_start();

include("functions/functions.php");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Travel Bird : Home</title>

<link rel="stylesheet" href="styles/style.css" media="all">

<style>

.adminbtn {

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 1px 2px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 14px;

cursor: pointer;

-webkit-transition-duration: 0.4s; /\* Safari \*/

transition-duration: 0.4s;

float: right;

margin-top: 12px;

margin-left: 2px;

}

.adminbtn:hover {

box-shadow: 0 12px 16px 0 rgba(0, 0, 0, 0.24), 0 17px 50px 0 rgba(0, 0, 0, 0.19);

}

</style>

</head>

<body>

<!--Main container starts-->

<div class="main\_wrapper">

<!--Header starts-->

<?php include 'includes/header.php'; ?>

<!--Header ends-->

<!--Navbar starts-->

<?php include 'includes/navbar.php'; ?>

<!--Navbar ends-->

<!--Content starts-->

<div class="content\_wrapper">

<!--left-sidebar starts-->

<?php include "includes/left-sidebar.php"; ?>

<!--left-sidebar ends-->

<div id="content\_area">

<?php cart(); ?>

<div id="shopping\_cart">

<span style="float: right;font-size: 18px;padding: 5px;line-height: 40px;">

<?php

if (isset($\_SESSION['customer\_email'])) {

echo "<b>Welcome: </b>" . $\_SESSION['customer\_email'] . "<b style='color: yellow;'> Your</b>";

} else {

echo "<b>Welcome Guest:</b>";

}

?>

<b style="color: yellow;">Shopping Cart-</b> Total Items: <?php total\_items(); ?>

Total Price: <?php total\_price(); ?> <a href="cart.php" style="color: yellow;">Go to Cart</a>

<?php

if (!isset($\_SESSION['customer\_email'])) {

echo "<a href='checkout.php' style='color: orange;'>User Login</a>";

} else {

echo "<a href='logout.php' style='color: orange;''>Logout</a>";

}

?>

<button class="adminbtn"><a style="text-decoration: none; color: #ffffff;"

href="admin\_area/index.php">Admin Login</a></button>

</span>

</div>

<div id="packages\_box">

<?php getPack(); ?>

<?php getCatPack(); ?>

<?php getTypePack(); ?>

</div>

</div>

</div>

<!--Content ends-->

<!--footer starts-->

<?php include "includes/footer.php";?>

<!--footer ends-->

</div>

<!--Main container ends-->

</body>

</html>

<?php

session\_start();

include("functions/functions.php");

include("includes/db.php");

global $con;

if (isset($\_POST['register'])) {

$ip = getIp();

$c\_name = $\_POST['c\_name'];

$c\_email = $\_POST['c\_email'];

$c\_pass = $\_POST['c\_pass'];

$c\_passport = $\_POST['c\_passport'];

$c\_image = $\_FILES['c\_image']['name'];

$c\_image\_tmp = $\_FILES['c\_image']['tmp\_name'];

$c\_country = $\_POST['c\_country'];

$c\_city = $\_POST['c\_city'];

$c\_contact = $\_POST['c\_contact'];

$c\_address = $\_POST['c\_address'];

// image will upload there

move\_uploaded\_file($c\_image\_tmp, "customer/customer\_images/$c\_image");

$insert\_c = "INSERT INTO customers (customer\_ip,customer\_name,customer\_email,customer\_pass,c\_passport,customer\_country,customer\_city,customer\_contact,customer\_address,customer\_image) VALUES ('$ip','$c\_name','$c\_email','$c\_pass','$c\_passport','$c\_country','$c\_city','$c\_contact','$c\_address','$c\_image')";

$run\_c = mysqli\_query($con, $insert\_c);

$sel\_cart = "SELECT \* FROM cart WHERE ip\_add='$ip'";

$run\_cart = mysqli\_query($con, $sel\_cart);

$check\_cart = mysqli\_num\_rows($run\_cart);

if ($check\_cart == 0) {

$\_SESSION['customer\_email'] = $c\_email;

echo "<script>alert('Account has been created successfully. Thanks!')</script>";

echo "<script>window.open('customer/my\_account.php','\_self')</script>";

} else {

$\_SESSION['customer\_email'] = $c\_email;

echo "<script>alert('Account has been created successfully. Thanks!')</script>";

echo "<script>window.open('checkout.php','\_self')</script>";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Travel Bird : Register</title>

<link rel="stylesheet" href="styles/style.css" media="all">

<style type="text/css">

#fixm {

font-size: 15px;

padding: 4px;

font-family: arial;

}

#fixi {

font-size: 14px;

font-family: arial;

}

#btn {

font-family: arial;

font-size: 14px;

background-color: #4CAF50; /\* Green \*/

border: none;

color: white;

padding: 10px 10px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

transition-duration: 0.4s;

cursor: pointer;

background-color: #32E875;

color: black;

}

#btn:hover {

background-color: #4CAF50;

color: white;

}

</style>

</head>

<body>

<!--Main container starts here-->

<div class="main\_wrapper">

<!--Header starts here-->

<?php include 'includes/header.php'; ?>

<!--Header ends here-->

<!--Navbar starts here-->

<?php include 'includes/navbar.php'; ?>

<!--Navbar ends here-->

<!--Content starts here-->

<div class="content\_wrapper">

<!--left-sidebar starts-->

<?php include "includes/left-sidebar.php"; ?>

<!--left-sidebar ends-->

<div id="content\_area">

<?php cart(); ?>

<div id="shopping\_cart">

<span style="float: right;font-size: 18px;padding: 5px;line-height: 40px;">Welcome Guest! <b

style="color: yellow;">Shopping Cart-</b> Total Items: <?php total\_items(); ?> Total Price: <?php total\_price(); ?> <a

href="cart.php" style="color: yellow;">Go to Cart</a></b></span>

</div>

<form action="customer\_register.php" method="post" enctype="multipart/form-data">

<table align="center" width="750" style="margin-top: 20px;">

<tr align="center">

<td colspan="6">

<h2 style="margin-bottom: 15px; font-family: Cambria;">Create an Account</h2>

</td>

</tr>

<tr>

<td align="right" id="fixm">Your Name:</td>

<td><input id="fixi" type="text" name="c\_name" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Email:</td>

<td><input type="email" name="c\_email" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Password:</td>

<td><input id="fixi" type="password" name="c\_pass" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Passport ID:</td>

<td><input type="text" name="c\_passport" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Image:</td>

<td><input id="fixi" type="file" name="c\_image" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Country:</td>

<td>

<select name="c\_country" required="">Select a

<option>Bangladesh</option>

<option>India</option>

<option>Japan</option>

<option>China</option>

<option>Russia</option>

<option>Portugal</option>

<option>England</option>

<option>Brazil</option>

<option>Spain</option>

<option>France</option>

<option>Switzerland</option>

<option>Croatia</option>

<option>Argentina</option>

</select>

</td>

</tr>

<tr>

<td align="right" id="fixm">Your City:</td>

<td><input id="fixi" type="text" name="c\_city" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Contact:</td>

<td><input id="fixi" type="text" name="c\_contact" required=""></td>

</tr>

<tr>

<td align="right" id="fixm">Your Address:</td>

<td><input id="fixi" type="text" name="c\_address" required=""></td>

</tr>

<tr align="center">

<td colspan="6"><input id="btn" type="submit" name="register" value="Create Account">

</td>

</tr>

</table>

</form>

</div>

</div>

<!--Content ends here-->

<!--footer starts-->

<?php include "includes/footer.php"; ?>

<!--footer ends-->

</div>

<!--Main container ends here-->

</body>

</html>

**5- Work schedule (find attached, change the dates according to your project work)**

A screenshot of a group of people

AI-generated content may be incorrect.

A screenshot of a white background

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.