**Modified Database**

**Database Name: sp\_travel**

**users**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **user\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **username** | VARCHAR(255) | Not Null, Unique |
| **email** | VARCHAR(255) | Not Null, Unique |
| **password** | VARCHAR(3000) | Not Null |
| **profile\_pic\_url** | VARCHAR(3000) | Not Null |
| **role** | VARCHAR(255) | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**travels**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **travel\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **title** | VARCHAR(255) | Not Null, Unique |
| **description** | VARCHAR(3000) | Not Null |
| **price** | DOUBLE | Not Null |
| **country** | VARCHAR(255) | Not Null |
| **travel\_period** | VARCHAR(255) | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**itineraries**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **itinerary\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_travel\_id** | INT | Not Null, On Delete Cascade |
| **day** | INT | Not Null |
| **activity** | VARCHAR(3000) | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**reviews**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **review\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_user\_id** | INT | Not Null, On Delete Cascade |
| **fk\_travel\_id** | INT | Not Null, On Delete Cascade |
| **content** | VARCHAR(3000) | Not Null |
| **rating** | DOUBLE | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**promotions**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **promotion\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_travel\_id** | INT | Not Null, On Delete Cascade |
| **promotion\_start** | DATETIME | Not Null |
| **promotion\_end** | DATETIME | Not Null |
| **discount\_percentage** | DOUBLE | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**images**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **image\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_travel\_id** | INT | Not Null, On Delete Cascade |
| **image\_path** | VARCHAR(255) | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**carts**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **cart\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_user\_id** | INT | Not Null |
| **fk\_travel\_id** | INT | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

**purchases**

|  |  |  |
| --- | --- | --- |
| **Columns** | **Datatypes** | **Remarks** |
| **purchase\_id (Primary Key)** | INT | Not Null, Auto Increment |
| **fk\_user\_id** | INT | Not Null |
| **fk\_travel\_id** | INT | Not Null |
| **created\_at** | DATETIME | Not Null, Default Now |

Arrows point from foreign keys to referenced primary keys in other tables.

The *user\_id* columns contain the IDs of the users. In the **users** table, it is the primary key. In other tables, *fk\_user\_id* are foreign keys that references the **users** table's *user\_id* primary keycolumn. All the foreign keys are set with ON DELETE CASCADE so corresponding rows in their tables would be deleted when a row in **users** table is deleted.

The *travel\_id* columns contain the IDs of the travel listings. In the **travels** table, it is the primary key. In other tables, *fk\_travel\_id* are foreign keys that references the **travels** table's *travel\_id* primary keycolumn. All the foreign keys are set with ON DELETE CASCADE so corresponding rows in their tables would be deleted when a row in **travels** table is deleted.

*fk\_user\_id* and *fk\_travel\_id* in the **reviews** and **carts** table are made to be a multi-column unique key *user\_travel\_same* (connected by red line), to make sure that each user can only leave one review for each travel listing, or book only one of each travel listing at a time.

Images are stored in the file system itself, then its path in the file system is stored as a reference in the MySQL database. *fk\_travel\_id* and *image\_path* in the **images** table are made to be a multi-column unique key *travel\_image\_path* (connected by red line), to make sure that each image has a unique path in the file system.

**Changes**

* "role" and "password" columns added to "users" table, to store the role and password hash of each user.
* Created "carts" table, to hold the shopping carts data of users.
* Created "purchases" table, to hold the purchase history data of users.

**Modified API Endpoints**

**Modified API Endpoints**

**1. GET /users/**

Additional step to remove password hashes of users from the users' information returned by the MySQL database before sending it as a response.

**2. GET /users/:user\_id**

Additional step to remove password hash of user from the user's information returned by the MySQL database before sending it as a response.

**3. POST /users/**

The model function and its SQL query have been modified so that it can take a role and password values to insert into the "users" table when creating a new user. The password will also be hashed with SHA-224 algorithm in NodeJS before it is inserted.

**4. PUT /users/:user\_id**

The model function and its SQL query have been modified so that it can take a role and password values to update the "users" table when updating a user. The password will also be hashed with SHA-224 algorithm in NodeJS before it is updated.

**5. Others**

Some endpoints have middleware added where appropriate, to check certain things about the users making the requests, such as whether they are logged in, their identity is valid, or are an admin. The middleware then acts according to the results of those checks.

**Added API Endpoints**

**1. GET /admin/verify/**

Solely used to check if user is admin without returning anything from the database. Does not interact with the database, so does not use functions at model level.

**2. POST /login/**

Used to verify user's identity when logging in. Uses "users.verifyUser" model function, also created for this assignment.

**3. GET /travels/last/**

Used to get last travel listing added. Uses "travels.getTravelLast" model function, also created for this assignment.

**4. GET /travels/search/:country/:price/:travel\_period/**

Used to get travel listings matching given search criteria. Uses "travels.getTravelsSearch" model function, also created for this assignment.

**5. GET /travels/:travel\_id/**

Used to get information of one travel listing. Uses "travels.getTravel" model function.

**6. GET /travels/:fk\_travel\_id/images/showcase/**

Used to get the showcase image of travel listings. Uses "images.getImageShowcase" model function, also created for this assignment.

**7. GET /users/:fk\_user\_id/carts/**

Used to get a user's shopping cart of booked travel listings. Uses "carts.getCart" model function, also created for this assignment.

**8. POST /users/:fk\_user\_id/travels/:fk\_travel\_id/carts/**

Used to add a booked travel listing to a user's shopping cart. Uses "carts.addCart" model function, also created for this assignment.

**9. DELETE /users/:fk\_user\_id/carts/**

Used to remove a booked travel listing from a user's shopping cart. Uses "carts.delCart" model function, also created for this assignment.

**10. POST /users/:fk\_user\_id/travels/:fk\_travel\_id/purchases/**

Used to add purchased travel listings after checking out with shopping cart to user's purchase history. Uses "purchases.addPurchase" model function, also created for this assignment.