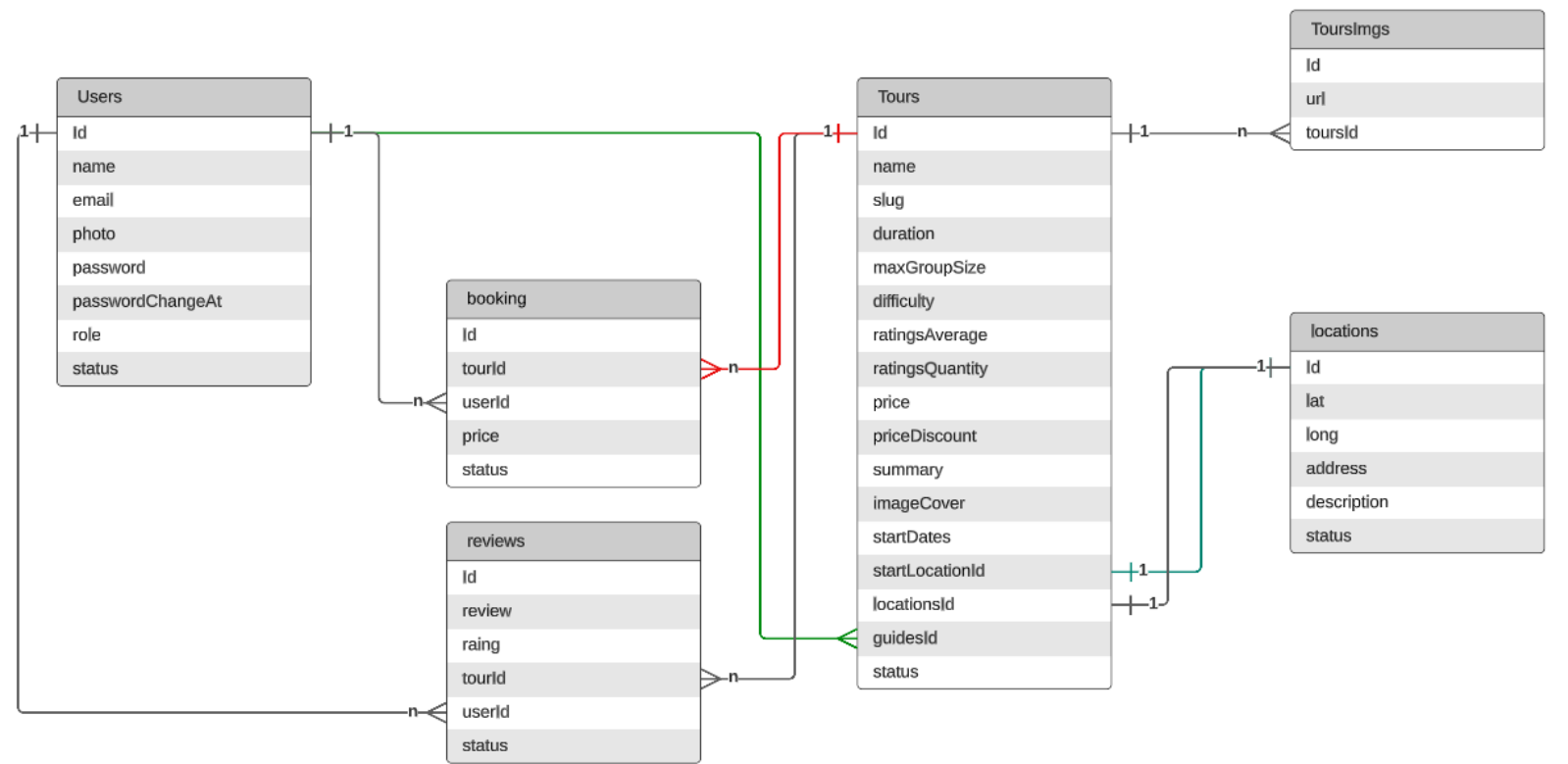


ACADEMLO TOURS



Estudiante de Academlo.

Espero que este mensaje te encuentre bien. Me complace comunicarme contigo hoy para discutir un emocionante proyecto en el que me gustaría que te involucraras. Como bien sabes, nuestra empresa, ExploraVenture Tours, se dedica a organizar y ofrecer emocionantes tours a nuestros clientes.

Como propietario de la empresa, estoy buscando llevar nuestra experiencia de tours al siguiente nivel, y para ello necesitamos una aplicación web que nos permita gestionar de manera eficiente todas nuestras operaciones y brindar una experiencia excepcional a nuestros clientes. Estoy convencido de que tu experiencia y habilidades como programador son esenciales para llevar a cabo este proyecto exitosamente.

Permíteme compartir contigo algunos detalles sobre esta aplicación que deseamos desarrollar. La aplicación debe permitir a nuestros clientes explorar los diferentes tours que ofrecemos, ver detalles como la ubicación de inicio y destino, las imágenes relacionadas, así como las reseñas de otros usuarios. Además, los usuarios deben poder registrarse, iniciar sesión y reservar tours específicos en fechas disponibles. También necesitaremos un sistema para gestionar las reservas, rastrear los guías asignados a cada tour y administrar las imágenes asociadas a cada uno.

Además, deseo destacar la importancia de la calidad y la atención al detalle en este proyecto. Queremos que la aplicación sea intuitiva, atractiva y fácil de usar para nuestros clientes. La usabilidad y la experiencia del usuario son fundamentales para que nuestros clientes se sientan cómodos al explorar y reservar nuestros tours. Estoy seguro de que, con tu conocimiento y habilidades técnicas, podrás cumplir con estos requisitos y brindarnos una aplicación que nos distinga en el mercado.

Si estás interesado en asumir este desafío y unirme a nosotros en este emocionante proyecto, me encantaría discutir los detalles más a fondo. Podemos programar una reunión para hablar sobre los requisitos adicionales, el cronograma y cualquier otra pregunta que puedas tener. Estoy abierto a tus ideas y sugerencias, ya que valoro tu experiencia y compromiso con la calidad.

Agradezco de antemano tu dedicación y entusiasmo para hacer de este proyecto una realidad. Espero con ansias trabajar juntos y lograr grandes resultados. ¡Hagamos que nuestros clientes disfruten de una experiencia de tours excepcional con nuestra nueva aplicación!

Saludos cordiales,

Luis Miguel Avendaño Lozano

ExploraVenture Tours

ENDPOINTS

/api/v1/auth		
HTTP VERB	ROUTE	DESCRIPTION
POST	/signup	Crear usuario, enviar por la req.body como formData: name, email, photo, password
POST	/login	Iniciar sesión, enviar por la req.body: email, password
GET	/renew	Renovar el token

- Solo la ruta /renew debe estar protegida por un método de autenticación, es decir, por JWT
- Se debe usar express-validator para validar los campos,
- Se debe encriptar la contraseña usando bcrypt.js
- En el modelo del usuario se deberá definir que el status puede ser: active o inactive, dejar el valor por defecto como active
- En el modelo del usuario se deberá definir que el rol puede ser: user, guide, admin dejar como valor por defecto, user
- Utilizar multer y firebase u otro servicio de nube para almacenar las imágenes.

/api/v1/user		
HTTP VERB	ROUTE	DESCRIPTION
GET	/	Debe listar todos los usuarios creados con status true
PATCH	/password	Debe permitir al usuario cambiar la contraseña, se debe enviar por la req.body la contraseña antigua y la nueva contraseña.
GET	/:id	Debe buscar un usuario con status true dado un id
PATCH	/:id	Debe poder actualizar el name y el email, enviar la información por la req.body
GET	/tours/:id	Se debe poder listar todos los usuarios que tengan rol “user” asociados a un tour, siendo el :id es el id del tour
GET	/:userId/tours/:id	Se debe poder buscar un usuario asociado a un tour, siendo el :userId el id del usuario y el :id el id del tour en el cual se desea buscar ese usuario.
DELETE	/:id	Debe poder eliminar la cuenta del usuario, se debe cambiar su status a inactive
GET	/bookings	Debe poder traer todas las reservas del usuario con estatus pending
GET	/bookings/:id	Debe poder traer una reserva en concreto, siendo :id el id de la reserva que se desea revisar

- Todas las rutas deben estar protegidas por un medio de autenticación, es decir, por JWT.
- Para las rutas patch /password /:id, y para la ruta delete /:id, solo el usuario dueño de la cuenta podrá ejecutar esta acción.
- Para las rutas get /tours/:id y /:userId/tours/:id solo los usuarios con rol: guide y admin podrán ejecutar esta acción,
- Las rutas patch deben contar con validaciones utilizando express-validator
- Las rutas get /bookings y /bookings/:id debe buscar las reservas del usuario en sesión (del token que se envía).
- Para las rutas get /bookings y /bookings/:id se debe incluir la siguiente información:
  - Información del tour al cual se hizo la reserva
  - Información de los guías asociados al tour
  - Información del lugar donde inicia el tour, y de donde se va a hacer el tour.
- Los usuarios pueden tener status: active o inactive

/api/v1/tours		
HTTP VERB	ROUTE	DESCRIPTION
GET	/	Obtener todos los tours con status active, no es necesario incluir información de otras tablas
GET	/:id	Obtener un tour activo dado un id, siendo el :id el id del tour, se debe incluir la información del guía del tour, así como donde empieza el tour y donde será realizado el tour, y las reviews y fotos de dicho tour.
POST	/	Se debe poder crear un tour, se debe enviar por la req.body toda la información del modelo a excepción del id y el status.
PATCH	/:id	Se debe poder actualizar el tour dado un id, siendo :id el id del tour. Toda la información a excepción del id y el status puede actualizarse
DELETE	/:id	Se debe poder eliminar un tour, actualizando el status de tour a deleted.

- Todas las rutas deben estar protegidas por un medio de autenticación, es decir, por JWT.
- Las rutas POST y PATCH deben estar validadas por express-validator
- Todas las rutas excepto GET: / y /:id deben estar protegidas para que solo los usuarios con role admin puedan utilizarlas (validar el rol del usuario en sesión)
- Te en cuenta que para poder crear un tour es necesario enviar la información en formato form-data, esto porque debemos además registrar en la tabla de toursImgs las respectivas imágenes, y subirlas a un servicio de nube, como firebase o cloudinary entre otros, la elección es tuya.
- Los tours, pueden tener status: active o deleted, se recomienda crear en enum en su modelo.
- Utilizar multer para poder recibir imágenes en el backend.

/api/v1/reviews		
HTTP VERB	ROUTE	DESCRIPTION
GET	/	Obtener todas las reviews con status true
GET	/:id	Obtener una review con status true dado un id
POST	/:tourId	Se debe crear una review a un tour, siendo :tourId el id del tour al cual se le creara la review. Enviar review y rating por la req.body y el userId deberán obtenerlo del usuario en sesión.
PATCH	/:tourId/:id	Se debe poder actualizar una review, siendo el :tourId el id del tour, y :id el id de la review que se desea actualizar. Solo se actualizara la review y rating las cuales se deben proporcionar por la req.body.
DELETE	/:tourId/:id	Se debe poder eliminar una review, siendo el :tourId el id del tour, y :id el id de la review que se desea eliminar, se debera cambiar el status de la review a false.

- Todos los métodos deben estar validados con un método de autenticación es decir JWT
- Para los endpoints post y patch deben estar validados con express-validator
- Antes de actualizar o eliminar cualquier review verificar que el tour exista y que la review exista
- Solo el usuario dueño de la review podrá eliminar o actualizar su review.
- Antes de crear una review se debe validar que el tour al cual se le hará la review exista.
- Las reviews pueden tener status: true o false

/api/v1/booking		
HTTP VERB	ROUTE	DESCRIPTION
GET	/	Obtener todas las reservas realizadas por los usuarios, solo el usuario administrador podrá ejecutar esta acción, además se debe incluir la información del usuario que hizo la reserva y el tour el cual reservo.
GET	/:id	Obtener una reserva dado un id, solo el usuario dueño de la reserva podra ejecutar esta acción
POST	/	Se debe crear una reserva, enviar userId, tourId, y price por la req.body.
PATCH	/:id	Se debe poder actualizar una reserva, solo se podrá actualizar el precio, enviarlo por la req.body, solo el usuario que hizo la reserva puede modificarla.
DELETE	/:id	Se debe poder cancelar una reserva, modificar el status a cancelled, solo el usuario que hizo la reserva puede eliminarla.

- Todos los métodos deben estar validados con un método de autenticación es decir JWT
- Los endpoints post y patch deben estar validados con express-validator.
- Al crear una reserva debe enviarse un correo electrónico a la persona que hizo la reserva.
- Validar que antes de hacer una reserva el tour este disponible
- La reserva puede tener los siguientes status:
  - Pending: hizo una reserva pero el monto con el que aparto la reserva, no es el monto de lo que vale el tour, si no una parte. (este monto es el Price que esta en la tabla booking)
  - Completed: Cuando se hace la reserva y se cancela o paga todo el monto de lo que vale el tour el status es completed
  - Cancelled: cuando se cancela una reserva
- Si al crear una reserva se paga una parte del tour, se debe colocar status pending
- Si al crear una reserva se paga todo el monto del tour se debe colocar status completed
- En el caso de que existan dos tours diferentes con la misma fecha, solo se podrá reservar un tour.

El líder de desarrollo ha considerado que tienes las capacidades necesarias, para seguir haciendo los planteamientos que faltan, es por eso que te ha pedido, que hagas el planteamiento de los crud. Para la tabla locations, El confía en ti y reconoce tus habilidades, entonces deberas plantear los crud con sus respectivas validaciones y desarrollarlos.

También te informa, que la guía anterior es como su nombre lo indica una GUIA, el sabe que tienes las capacidades necesarias para saber si hay que mas validaciones, o incluso modificar alguna de las funcionalidades si así lo crees conveniente.