Patients

models -> PatientSchema holds the schema for the Patient records.

- A brief overview of the schema and the corresponding pre/post middleware methods is as follows:
    - Patient slug creation - pre
    - Creation of a location field and geocode - pre
    - Password encryption - pre
    - Sign and return JWT token - getSignedJwtToken
    - Password matching between user input and hashed password in database - match password
    - Generate and hash password token - getResetPasswordToken

controllers -> The controller methods used on patients are stored here.

- The controller methods are as follows:
    - getPatients
        - This method attempts to retreive the data of all of the patients from the database in the form of a JSON.
    - getDoctorsWorked
        - This method attempts to retreive the data of all of the doctors that have worked with a specific patient from the database in the form of a JSON.
    - getPatientAppointments
        - This method attempts to retrieve the data of all of the appointments that are associated with a specific patient from the database in the form of a JSON.
    - getPatient
        - This method attempts to retrieve the data of a specific patient from the database in the form of a JSON
    - createPatient
        - This method attempts to create new data for a specific patient within the database, and verifies that the patient has been created.
    - loginPatient
        - This method analyzes the login credentials provided by the user, and either rejects the user with an invalid credentials message, or authenticates them with a token and grants them access.
    - updatePatient
        - This method updates the data for a specific patient that is stored in the database.
    - deletePatient
        - This method removes the data for a specific patient from the database.

    - patientPhotoUpload
        - This method accepts a file for uploading, and ensures that the file fits the upload criteria such as being a photo and adhering to a file size limit. The method also provides functionality to allow the user to create a filename for the photo file.
    - updateInsurancePackage
        - This method allow the user to change their insurance package, and includes several functionalities, such as error handling if the user inputs and insurance package ID that does not match any active insurance packages.
    - getAllInsurancePackages
        - This method attempts to retreive the data for all of the insurance packages from the database.
    - sendTokenResponse
        - This method gets a token from the model and creates a cookie and a send response.

routes -> The API calls are all stored here.

- We have the below routes for Patient:
    - getPatients
        - This route is protected, and users must a doctor to access it.
    - getAllInsurancePackages
        - This route is unprotected, and anyone may access it.
    - getDoctorsWorked
        - This route is protected, and users must be a patient to access it
    - createPatient
        - This route is unprotected, and anyone may access it.
    - loginPatient
        - This route is unprotected, and anyone may access it.
    - getpatientAppointments
        - This route is protected, and users must be a patient to access it
    - getPatient
        - This route is protected, and users must be a doctor to access it
    - updatePatient
        - This route is protected, and users must be a patient to access it
    - deletePatient
        - This route is protected, and users must be a patient to access it
    - patientPhotoUpload
        - This route is protected, and users must be a patient to access it
    - updateInsurancePackage
        - This route is protected, and users must be a patient to access it