Documentation

RZ1109

Ruinan Zhang

CS-6083 Final Project 1&2

**1 Backend Database design**

**1. 1 Assumption about database**

(a) Whenever a new user register, it will require a unique email address. This means each distinct email address can only be associated to one account. The email address cannot be changed once signed up. After registering, user will login with email address and then log in with password.

(b) For each item posted by any user(seller), system will assign a (random)unique auction\_id to the item which associated with auction info posed by the user at the same time.

If an item was not sold in an auction, the user (seller) then will have to choose to re-sell the item, which will generate another unique auction\_id associate with the new auction.

(c)Same user(buyer) can offer bid more than one time when the auction is ongoing, But the time that makes bid is unique, which means no same user can make more than one bid at the same time.

(d)We also assume that the bid made by a user must larger than the previous bid. So when a user wants to make a bid for an item, he or she will see the latest bid from previous user and his/her bid must be larger than that amount

(e) Assume seller reputation and buyer reputation are both generated randomly when a user signed up.

**1.2 Relational schema design**

User (**user\_id,** email, user\_name, address, phone, register\_date, active\_status, last\_active, paypal\_account, seller\_rep, buyer\_rep, password)

Primary key: user\_id

Item (**item\_id,** ***seller\_id*,** category, condition, description, pic\_url)

Primary key: item\_id

Foreign key: seller\_id referencing user

Auction\_buyer\_record(**auction\_id, *item\_id*, buyer\_id**, bid**, bid\_time)**

Primary key: auction\_id, buyer\_id, bid\_time

Foreign key: item\_id referencing item, buyer\_id referencing user

Auction\_seller\_record **(auction\_id**, ***item\_id***, min\_accept\_price, auction\_start\_time, aution\_end\_time)

Primary key: auction\_id

Foreign key: item\_id referencing item

Auction\_result (**auction\_id**, ***item\_id***, ***winner\_id***, final\_res, pay\_or\_not)

Primary key: auction\_id

Foreign key: item\_id referencing item, winner\_id referencing user, auction\_id referencing auction\_seller

Like (**user\_id**, **item\_id)**

Primary key: user\_id, item\_id

Foreign key: user\_id referencing user, item\_id referencing item

Search**(user\_id, search\_time, *itme\_id*,** filter\_used**)**

Primary key: user\_id, search\_time

Foreign key: user\_id referencing user, item\_id referencing item

**1.3 create database**

**a) for table user :**

CREATE TABLE `User` (

`user\_id` int(11) NOT NULL,

`email` varchar(45) NOT NULL,

`user\_name` varchar(45) DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`phone` int(11) DEFAULT NULL,

`register\_date` datetime NOT NULL,

`active\_status` varchar(5) DEFAULT NULL,

`last\_active` datetime DEFAULT NULL,

`paypal\_account` varchar(45) DEFAULT NULL,

`seller\_rep` int(11) DEFAULT NULL,

`buyer\_rep` int(11) DEFAULT NULL,

`password` varchar(45) DEFAULT NULL,

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**b) for table item:**

CREATE TABLE `item` (

`item\_id` int(11) NOT NULL,

`seller\_id` int(11) NOT NULL,

`category` varchar(45) NOT NULL,

`Condition` varchar(45) NOT NULL,

`Description` varchar(60) NOT NULL,

`pic\_url` varchar(60) DEFAULT NULL,

PRIMARY KEY (`item\_id`)

FOREIGN KEY (`seller\_id`) REFERENCES `user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**c) for table Auction\_buyer\_record:**

**CREATE TABLE `auction\_buyer\_record` (**

`auction\_id` int(11) NOT NULL,

`item\_id` int(11) DEFAULT NULL,

`buyer\_id` int(11) NOT NULL,

`bid` int(11) NOT NULL,

`bit\_time` datetime NOT NULL,

PRIMARY KEY (`auction\_id`,`buyer\_id`,`bit\_time`)

FOREIGN KEY (`item\_id`) REFERENCES `item` (`item\_id`)

FOREIGN KEY (`buyer\_id`) REFERENCES` user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**d) for table Auction\_seller:**

CREATE TABLE `auction\_seller` (

`auction\_id` int(11) NOT NULL,

`item\_id` int(11) NOT NULL,

`min\_accept\_price` varchar(45) NOT NULL,

`auction\_start\_time` datetime NOT NULL,

`aution\_end\_time` varchar(45) NOT NULL,

PRIMARY KEY (`auction\_id`,`item\_id`)

FOREIGN KEY (`item\_id`) REFERENCES `item` (`item\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**e) for table Auction\_res:**

CREATE TABLE `auction\_res` (

`auction\_id` int(11) NOT NULL,

`item\_id` int(11) DEFAULT NULL,

`winner\_id` int(11) DEFAULT NULL,

`final\_res` int(11) unsigned DEFAULT NULL,

`pay\_or\_not` varchar(10) DEFAULT NULL,

PRIMARY KEY (`auction\_id`)

FOREIGN KEY (`item\_id`) REFERENCES `item` (`item\_id`)

FOREIGN KEY (`winner\_id`) REFERENCES` user` (`user\_id`)

FOREIGN KEY (`auction\_id`) REFERENCES` auction\_seller` (`action\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**f) for table Like:**

CREATE TABLE `like` (

`user\_id` int(11) NOT NULL,

`item\_id` int(11) NOT NULL,

PRIMARY KEY (`user\_id`,`item\_id`)

FOREIGN KEY (`item\_id`) REFERENCES `item` (`item\_id`)

FOREIGN KEY (`user\_id`) REFERENCES` user` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

**g) for table Search:**

**CREATE TABLE `searches` (**

`user\_id` int(11) NOT NULL,

`search\_time` datetime NOT NULL,

`itme\_id` int(11) NOT NULL,

`filter\_used` varchar(45) DEFAULT NULL,

PRIMARY KEY (`user\_id`,`search\_time`)

FOREIGN KEY (`item\_id`) REFERENCES `item` (`item\_id`)

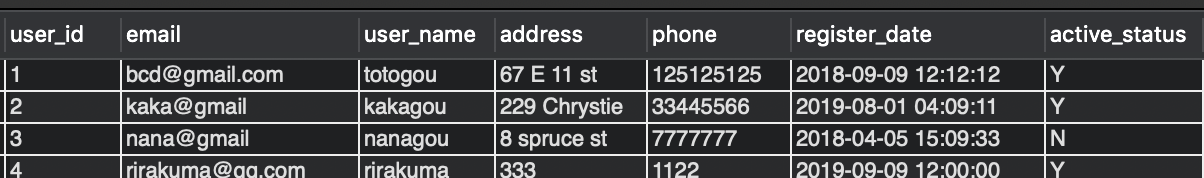
FOREIGN KEY (`user\_id`) REFERENCES` user` (`user\_id`)

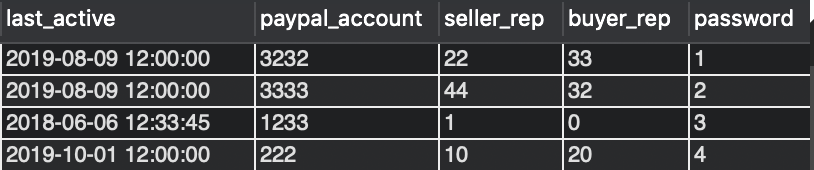
**) ENGINE=InnoDB DEFAULT CHARSET=utf8**

**1.4 Insert sample data**

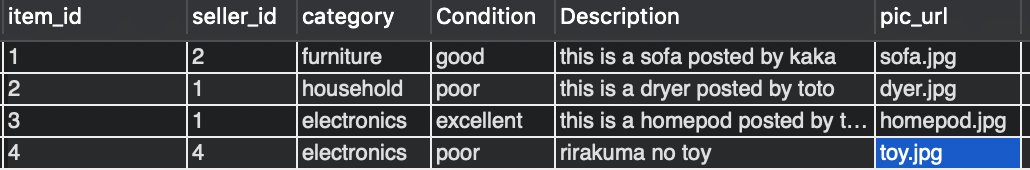
for the purpose of demonstration, below are some of sample data I inserted into my tables:

**a) for table user :**

****

****

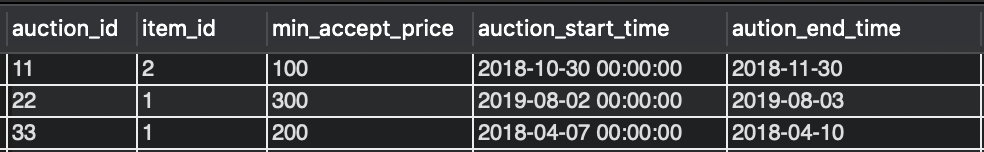
**b) for table item:**

****

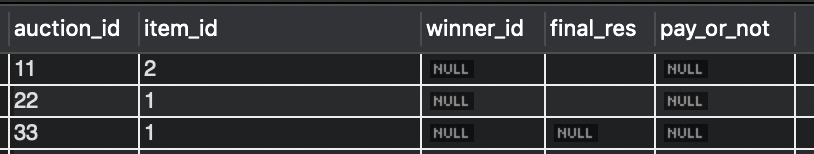
**c) for table Auction\_buyer\_record:**

****

**d) for table Auction\_seller:**



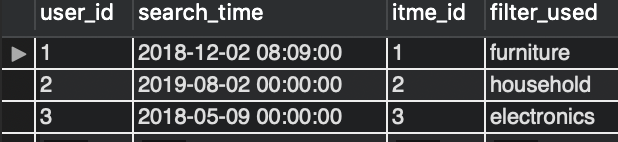
**e) for table Auction\_res:**

****

**f) for table Like:**



**g) for table Search:**



2.client and server side:

In this part, I used a local server on my computer using XAMPP, which include a local Apache server and MySQL server.

On the client side:

I used React.js as the framework for calling the endpoints and to load the data from back end.

Fetch data from the backend:

Use React alongside an Express App backend for wiring up the UI to fetch some data from the backend.

I started with create an express app with the express-generator utility;

Then created a react-backend folder for connection;

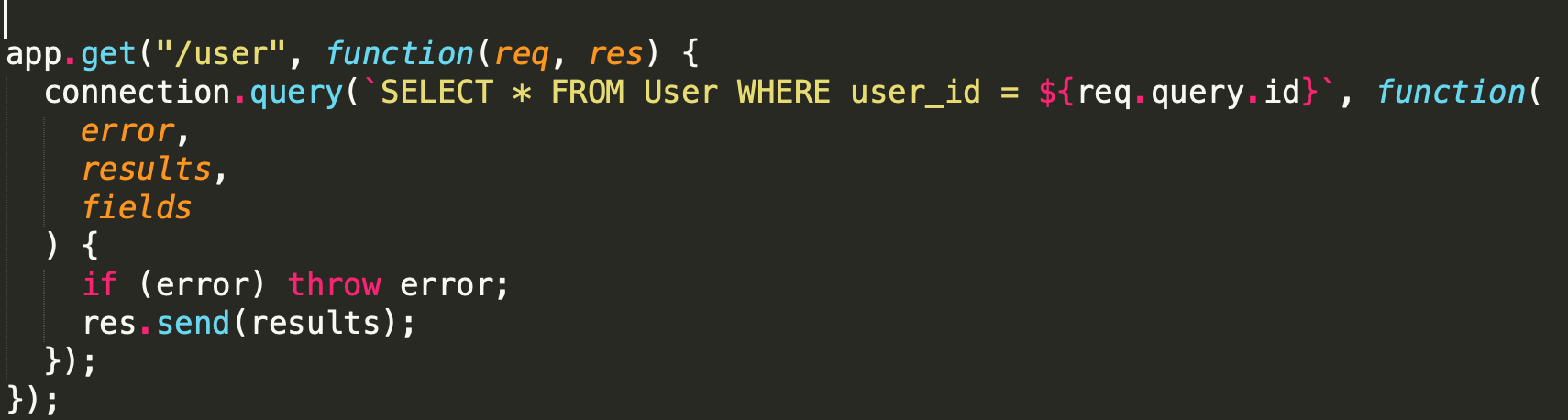
Then edited the react-backend/routes/users.js file as a simple way to return some data;

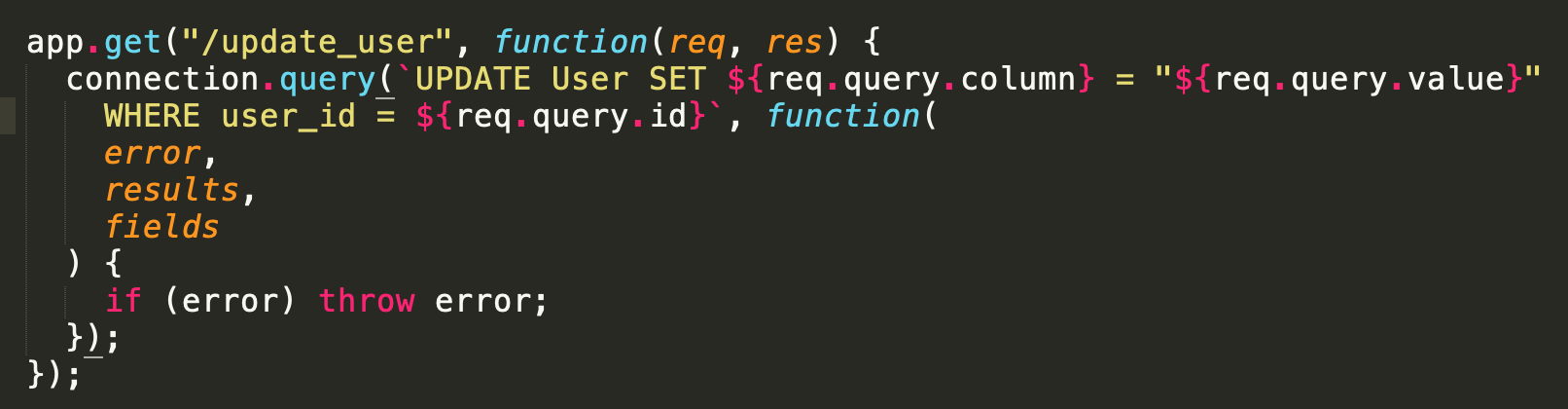
Every time I want to connect, I start up the app

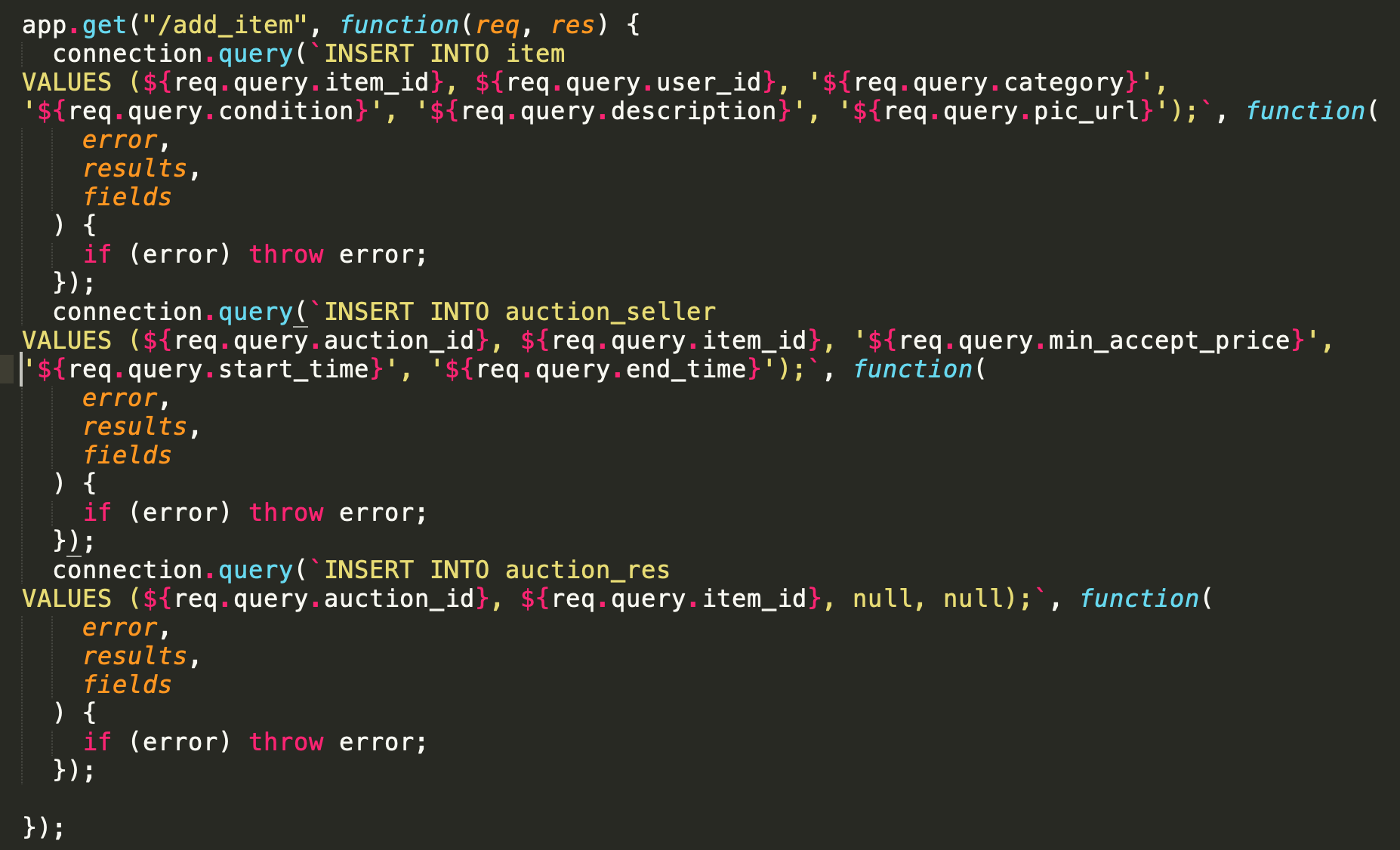
by running this:

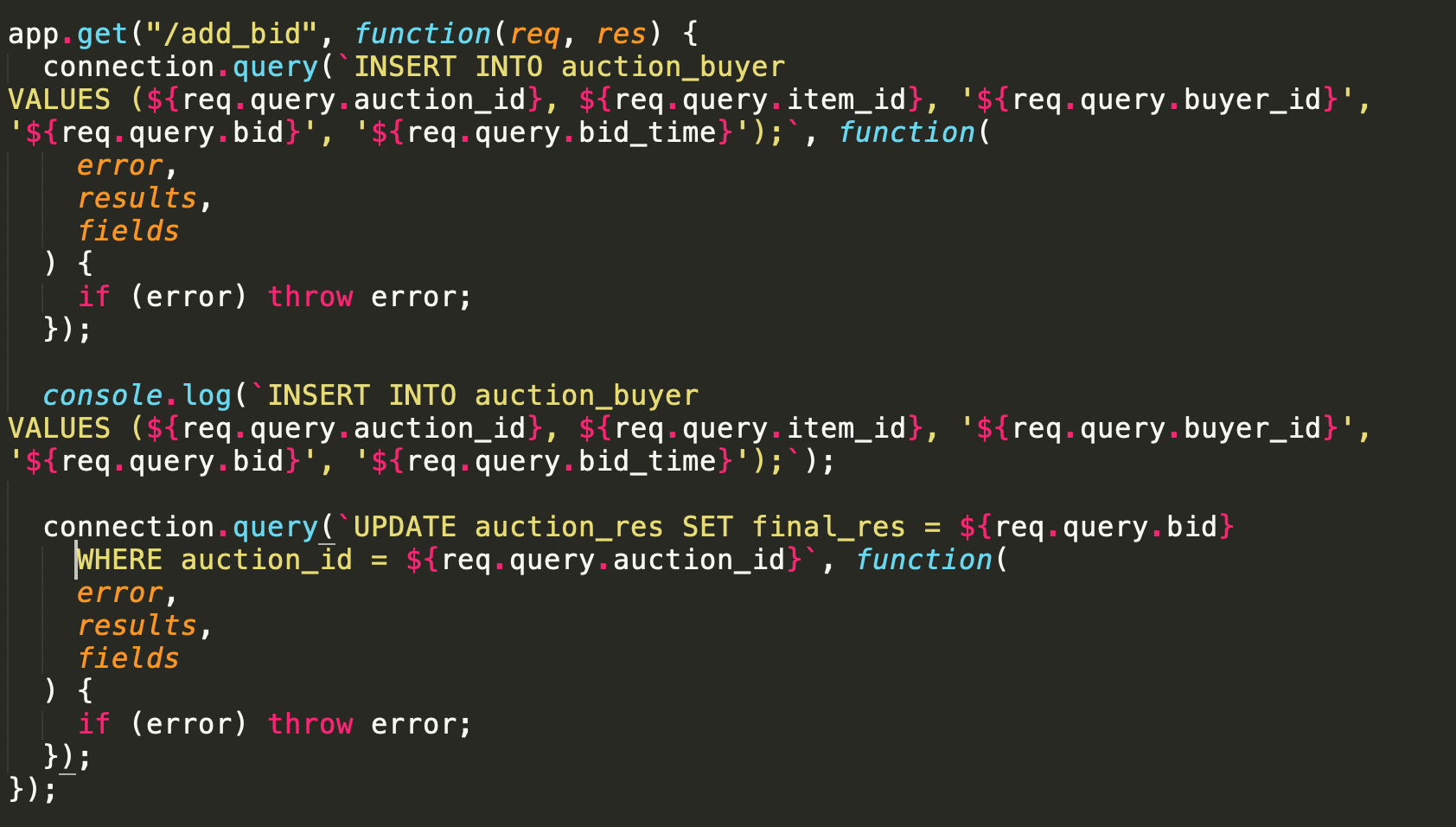
Here’s my sql query used in express.js to fetch data from the back-end:

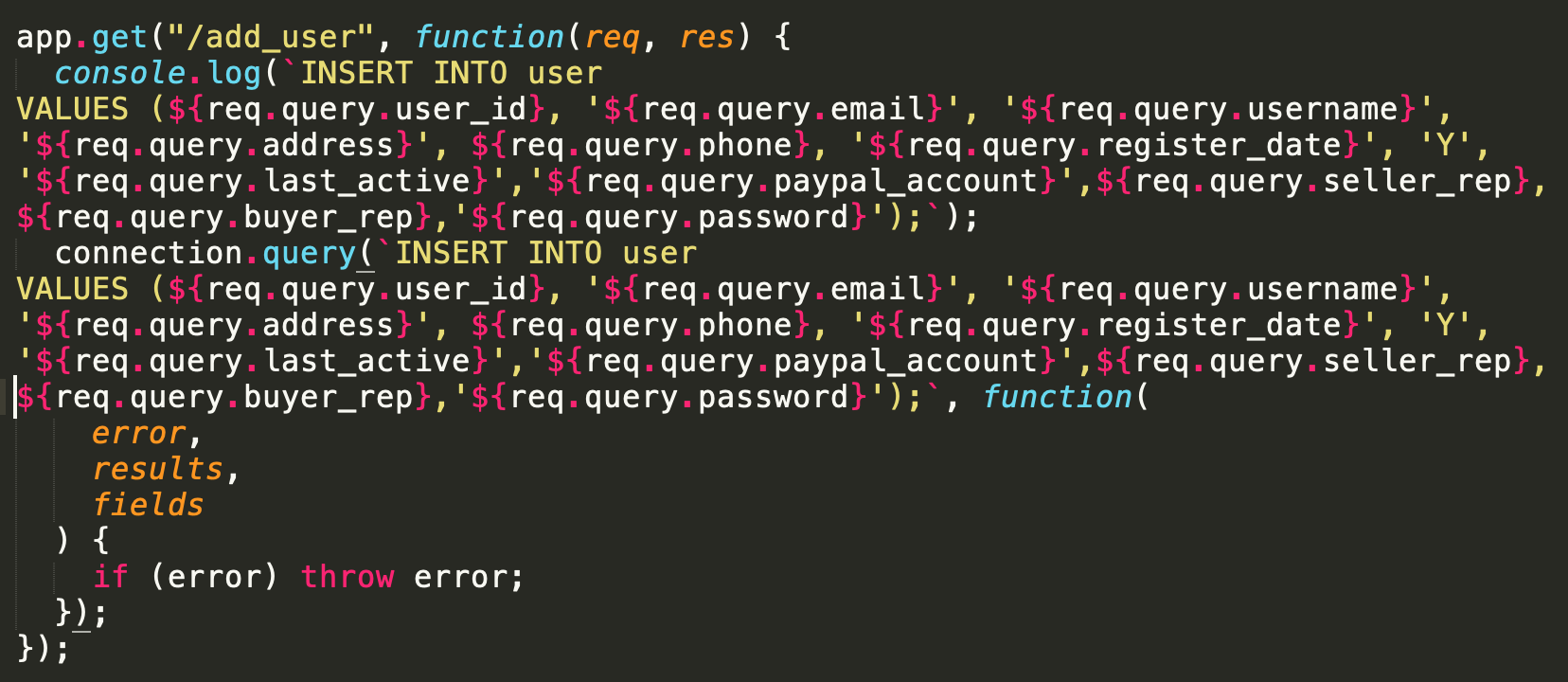
1. select user information with user\_id

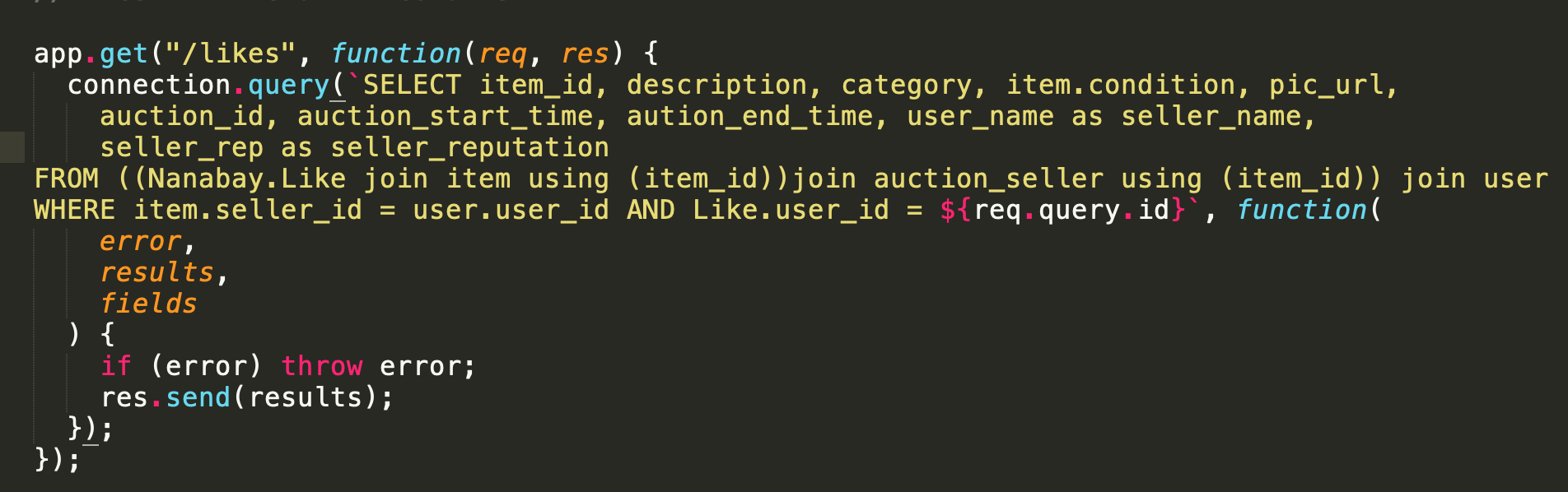


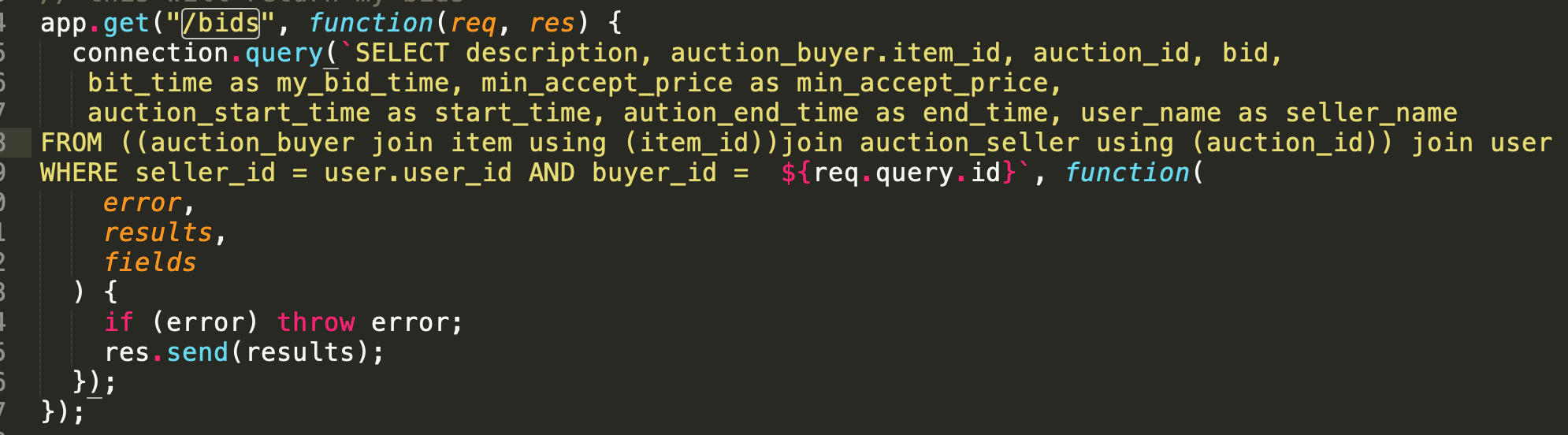
2. update a user’s info 

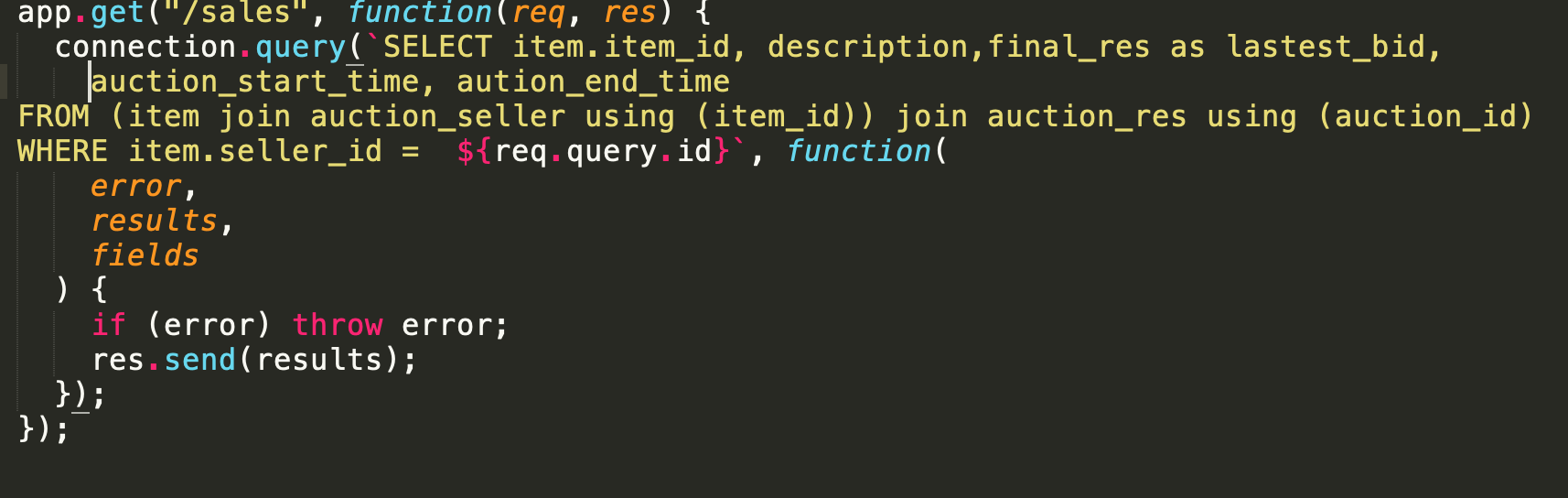
3. add an item to the database 

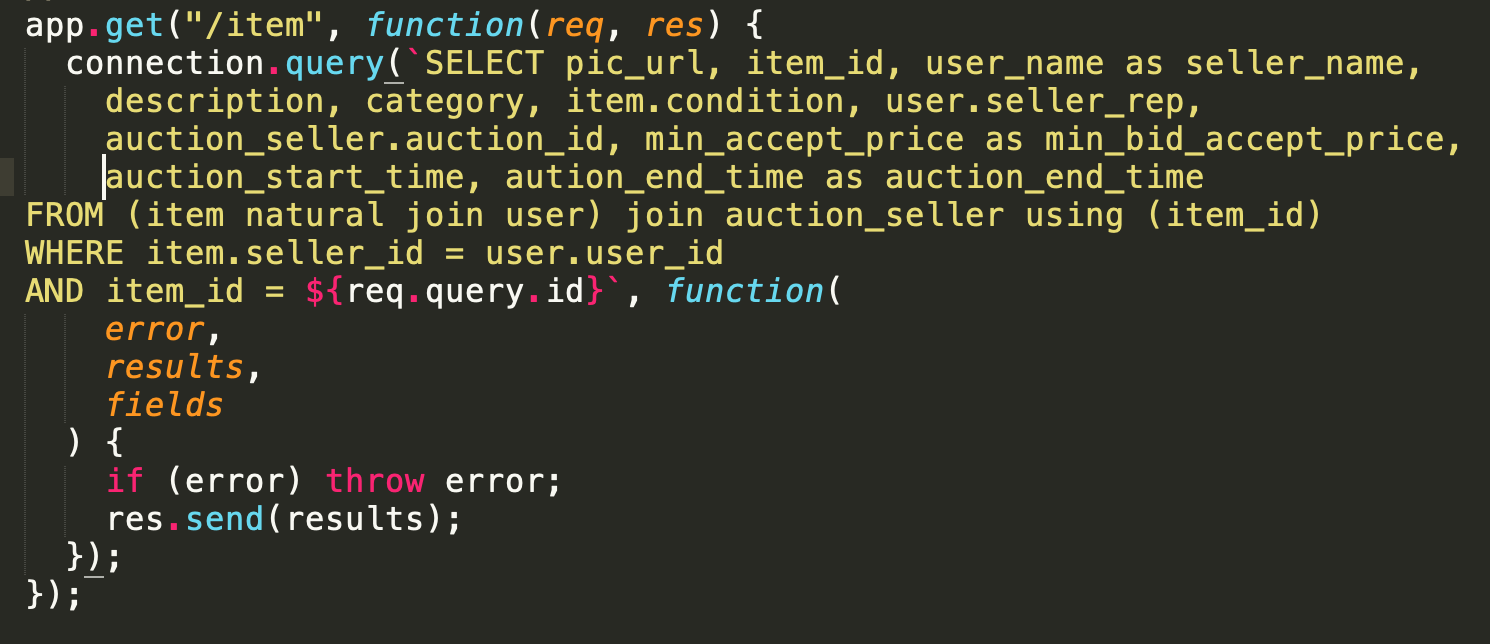
4. user add a bid to a specific item

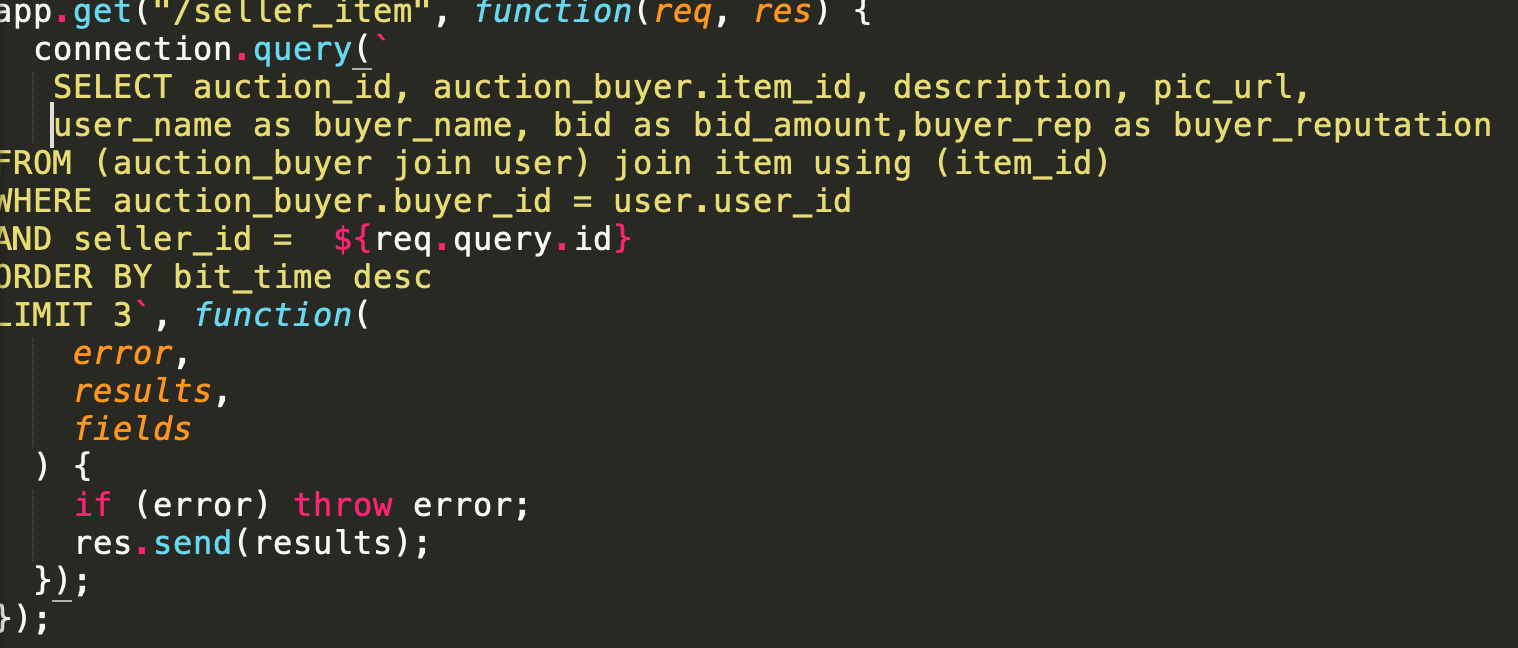
5. add user info to user table when a new user sign up 

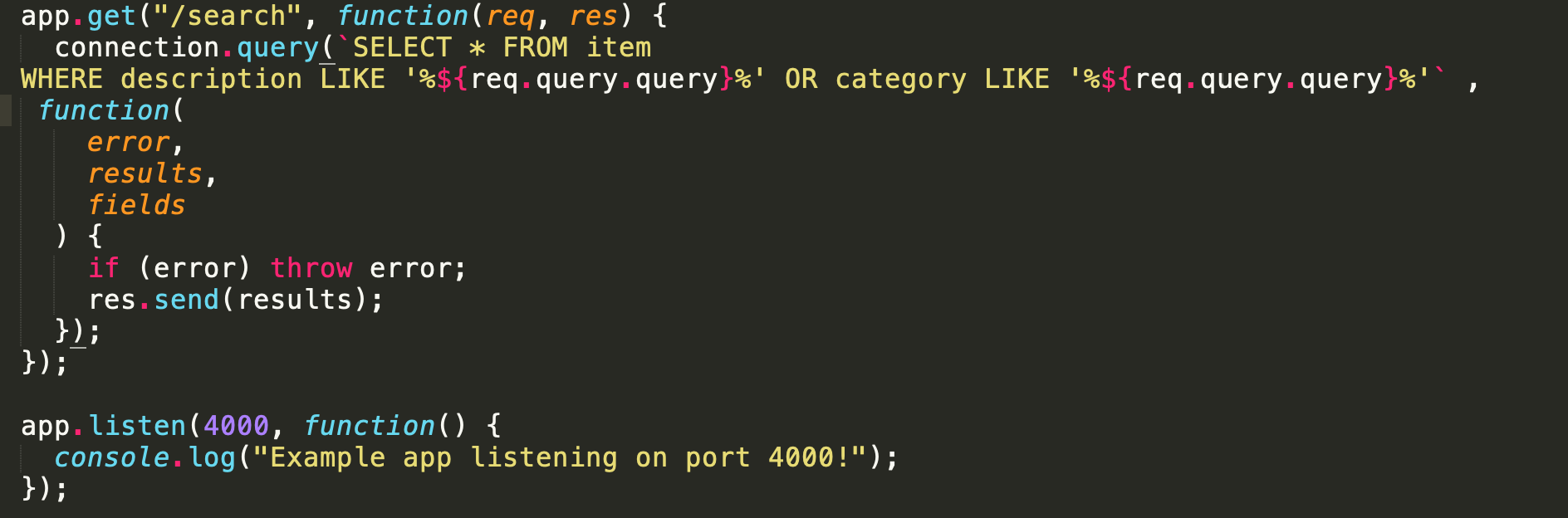
6. Find liked items for a sepecific user 

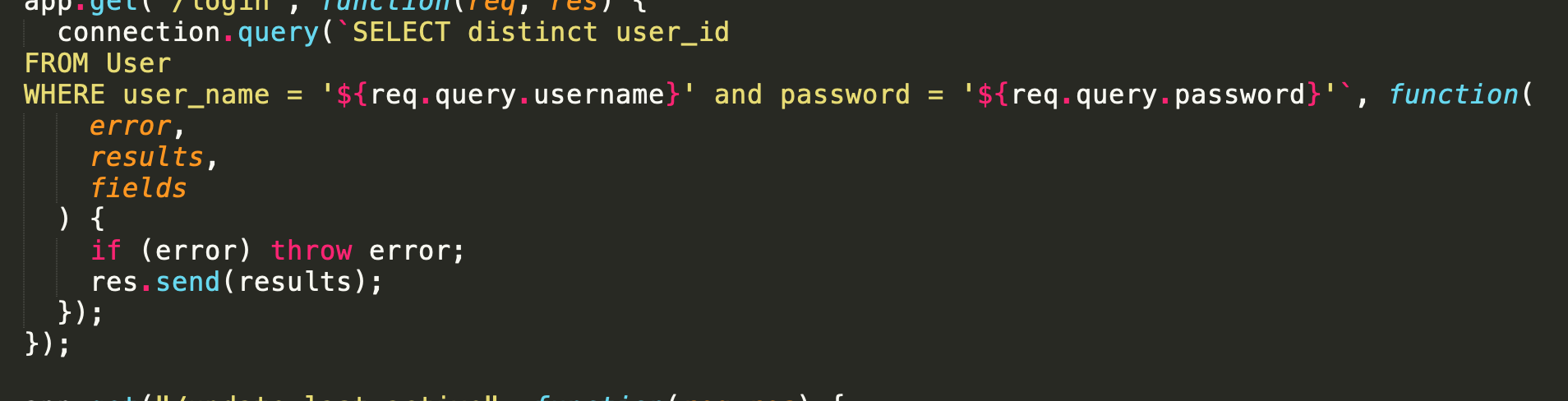
7. will find all bids made by a specific user 

8. will return all items a user sell or sold 

9. return all basic info about an item 

10. find latest three bids from all buyers for an item 

11. show search results for a specific user 

12. return user id if successfully log in, return null if wrong user\_name or password

13. update the last\_active time if a user loged in 