Brownbag

A restaurant search and review website

https://a5brownbag.herokuapp.com/

Team Information:

Full Name: *Jahandar Jamalli*
cdf id: c5jamall
Full Name: *Nguyen Quang Minh*
cdf id: g4minh
Full Name: *Fariz Huseynli*
cdf id: c4huseyn
Full Name: *Sajjad Siddiqui*
cdf id: g3sajjad

Detailed Design Description:

High level view:

We have several components in our application:

- User registration.
- User authentication including third party authentication Facebook.
- User / Restaurant profile
- Searching / Filtering in both user and admin sites.
- Restaurant recommendation system.
- Restaurant reviews, including commenting and rating.
- Messaging between users in the system.
- Admin account.

Description of each section of the system:

Database Models:

- Auth(_id, email, name, password, accountType)
- Avatar(_id, data)
- FBUser(_id, name, email, fbID, token, auth: Auth._id, age, preferredCuisine, avatar: Avatar._id)
- Message(_id, toId: Auth._email, FromId: Auth._email, timestamp, message)
- Restaurant(_id, name, location, cuisine, avatar: Avatar._id, auth: Auth._id, rating)
- Review(_id, userId : Auth._id, restaurantId : Auth._id, rating, comment)
- User(_id, name, age, preferredCuisine, avatar: Avatar._id, auth: Auth._id)

Registration component:

- Restaurant and user both sign up in the same page (/signup) but they have different sign up forms.
- There is a radio-box in (/signup) which the users can select to choose an appropriate form for them.
- There are variety of checks for the user sign up data in the front end, including email format, duplicated email, password length, name length, …
- There is a same set of checks applied in the server side, to cater for use who don't use the GUI to send data to the server.
- Depending on the form that the user chooses, our website will determine which account type to create for that user:
    - If the the user chooses to sign up as a regular user, then the very first user that signs up will be assigned an ADMIN account and any subsequent users will be considered as a REGULAR user. Both ADMIN and REGULAR profile data will be stored in User table

except for their email and password which are stored in Auth table for authentication purpose.

  - If the user chooses to sign up as a restaurant, then they are given a RESTAURANT account. The restaurant profile data will be stored in Restaurant table, except for their contact email and password which are stored in Auth table, also for the authentication purpose.
- Users who choose third party authentication as Facebook will have their account created the first time they use their Facebook account to login. We store their email in Auth table and their profile data which is retrieved from Facebook in the FBUser table.
- Restaurant, Regular and Facebook users who have successfully signed up will be directed to user main page at /users/main. The Admin instead will be directed to users/admin.

Login/ Logout/ Authentication:

- User can choose to login using their Facebook account or their local account.
- After successfully login, Admin user will be directed to /users/admin page. Restaurant, Regular, Facebook users will be directed to /users/main, just like when they sign up.
- Any users who are *not* an admin that try to access to the /users/admin will be directed back to the last page they viewed.
- Session is created for each user when they log in and destroyed after they log out. The Auth._id is saved in the user session when they log in and is used later to determine if they have the right to perform a particular task.
- For instance, a restaurant user may try to send a request to modify other user profile. The server will use Auth._id saved in the session to determine that is not an admin, thus refuse to perform the request.
- A user doesn’t have right to update other user profiles or password.
- Users who are not logged in (i.e. no Auth._id saved in their session) cannot access to /users/main or users/admin.

Searching/ Filtering:

- In admin page, /users/admin, the admin has a list of all users (including restaurant) available in the database and options to filter using the user account types.
- In user main page, /users/main, the user can see a list of all restaurants in the database. The user can filter the restaurants using their ratings, cuisines or both.
- There is a set of 10 cuisines to choose from and each restaurant is rated from 0 to 5 points.
- There are new restaurants that may not have any ratings yet, the users are given an option to list only the restaurants with rating.

Recommendation system:

- By clicking on one of the restaurant in the user main or in the admin page, the user is directed to the restaurant profile where they can view the restaurant profile.
- Along with profile information, the user will get a list of recommended restaurants, that offer at least one common cuisine with the restaurant that the user is viewing.
- Clicking on one the restaurant name in the recommend list will transfer the user to the corresponding restaurant profile page.

Restaurant review – rating and commenting system:

- In a restaurant profile page, the users whom account type are not RESTAURANT can give comments to the restaurant.
- Each review must have a rating, ranging from 0 to 5, but the comment is optional. Submitting a review with empty rating will result in the restaurant being rated 0/5.

- The list of customer reviews will be displayed in the restaurant profile page. A review without a comment won't be displayed, but the rating still count toward the average rating of the restaurant.
- A review when submitted, will be post to the Review table. Attaching to each review are the Auth._id of the sender and recipient.

Interaction between users – messaging system:

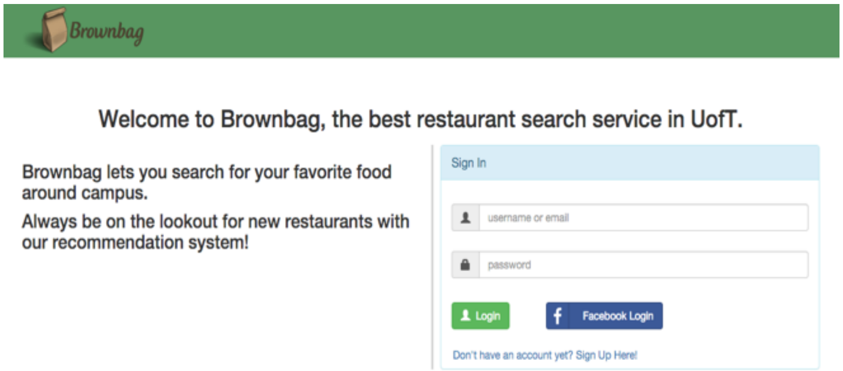
- In a restaurant profile page, the users are given an option to send messages to the restaurant.
- The admin can send messages to all other users in the system.
- In the profile page, the user who is recipient will have a list of usernames who wrote them some messages, if any. By clicking on one of those names, the user will be directed to the messaging interface where they can view and reply to the messages from that person.
- A message when submitted, will be post to the Message table. Attaching to each message are the email of the sender and recipient along with the message's time stamp.

Admin account:

- Just like in A4, the admin has right to edit other user profile, including user avatar but excluding the user password.
- The admin can also remove any other users in the database.
- They can send messages to any users in the database.
- Admin user is the very first user in the User table.
- Admin account when created cannot be removed.

List of pages and UI components:

/: login page, user can either login using FB account or using user in-site account.



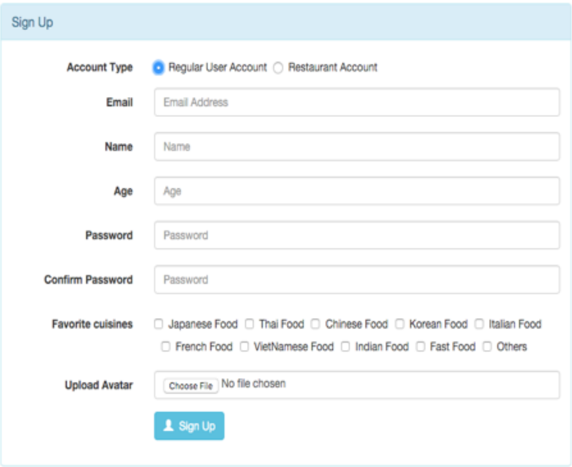
Welcome to Brownbag, the best restaurant search service in UofT.

Brownbag lets you search for your favorite food around campus.

Always be on the lookout for new restaurants with our recommendation system!

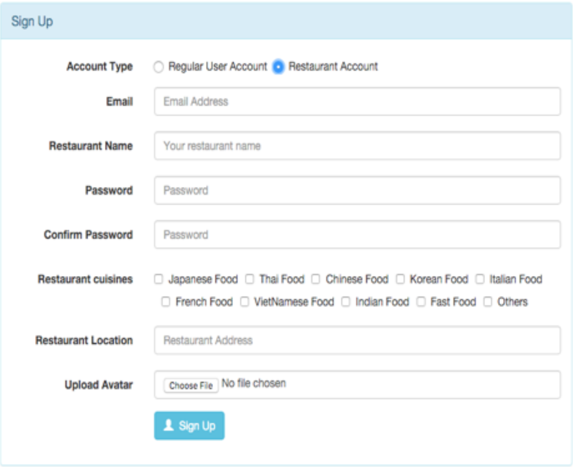
Don't have an account yet? Sign Up Here!

/signup: sign-up page, user can choose to sign up for a restaurant account or a regular user account.


Sign Up
Email
Email Address
Name
Name
Age
Age
Password
Password
Confirm Password
Password
Favorite cuisines
□ Japanese Food □ Thai Food □ Chinese Food □ Korean Food □ Italian Food
□ French Food □ VietNamese Food □ Indian Food □ Fast Food □ Others
Upload Avatar
Choose File
Sign Up

Account Type ● Regular User Account ○ Restaurant Account

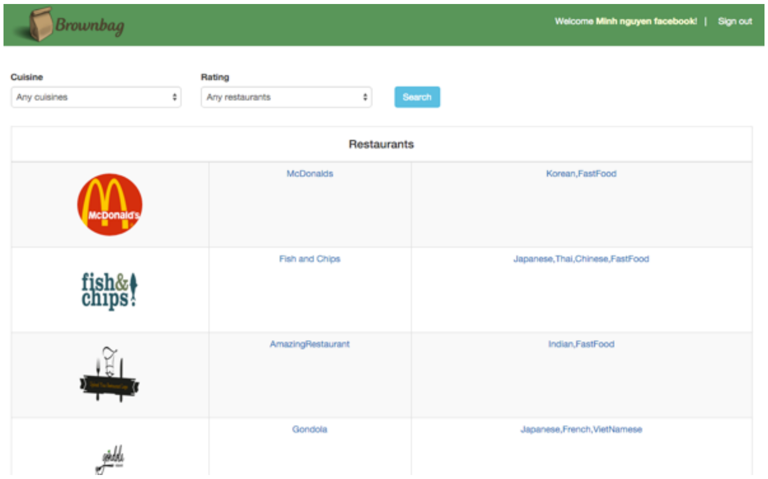
No file chosen



Account Type ○ Regular User Account ● Restaurant Account

No file chosen

/users/main: users main page, which has a list of restaurants in the database and the dropdown lists used to filter the restaurants based on their cuisine and average rating.



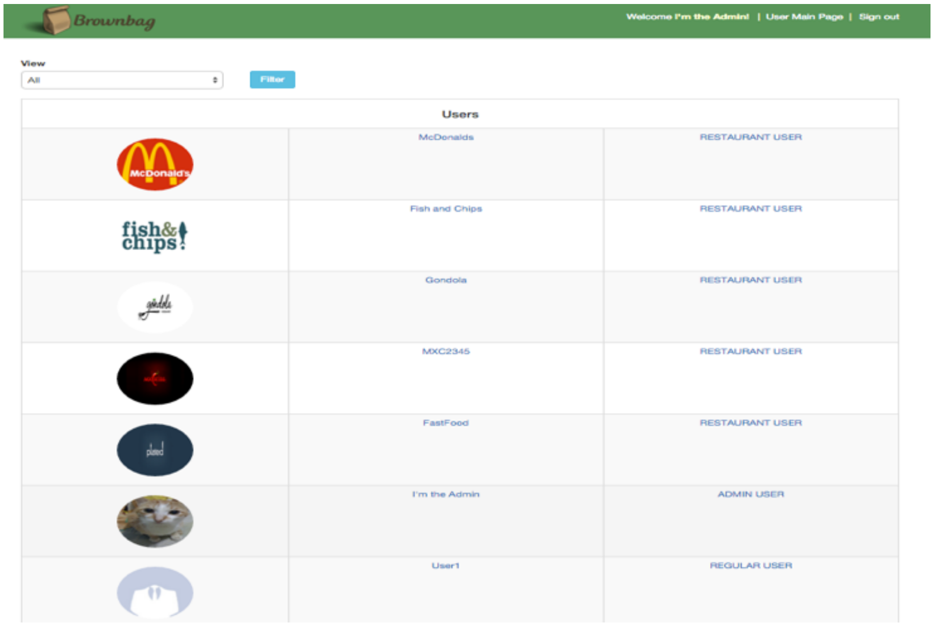
Cuisine: Any cuisines

Rating: Any restaurants

Restaurants

	McDonalds	Korean,FastFood
	Fish and Chips	Japanese,Thai,Chinese,FastFood
	AmazingRestaurant	Indian,FastFood
	Gondola	Japanese,French,VietNamese

/users/admin: admin main page, which has a list of all users in the database, including restaurant, regular user and FB users. The admin can use the drop down list to filter the users based on the user account type.



View: All

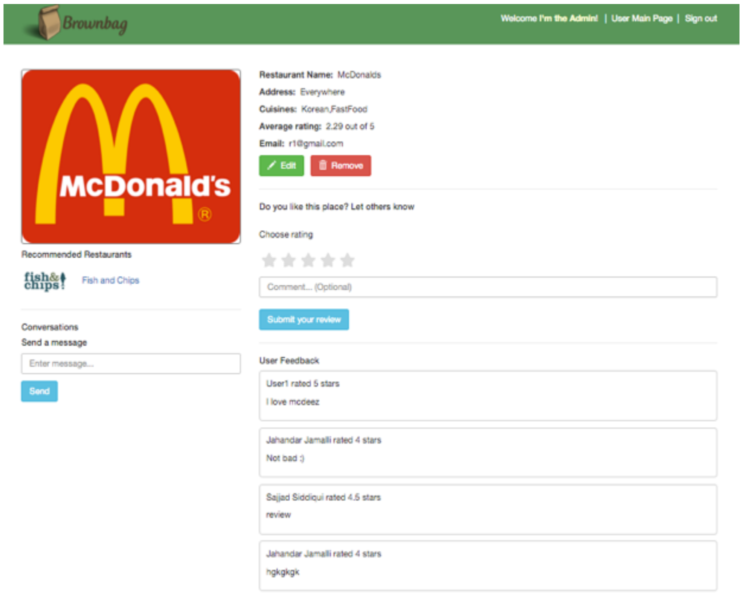
Users

	McDonalds	RESTAURANT USER
	Fish and Chips	RESTAURANT USER
	Gondola	RESTAURANT USER
	MXC2345	RESTAURANT USER
	FastFood	RESTAURANT USER
	I'm the Admin	ADMIN USER
	User1	REGULAR USER

/users/:id: user profile page, restaurant and regular users have different profile pages.

On the right is a restaurant profile page from the view of an **admin**, it has the following UI components:

- The restaurant logo
- List of recommend restaurant
- Conservation section for sending message.
- Restaurant profile information.
- The commenting section.
- A list of user feedbacks.



Restaurant Name: McDonalds
Address: Everywhere
Cuisines: Korean,FastFood
Average rating: 2.29 out of 5
Email: r1@gmail.com

Do you like this place? Let others know

Choose rating

☆☆☆☆☆

Comment... (Optional)

Recommended Restaurants

Fish and Chips

Conversations

Send a message

Enter message...

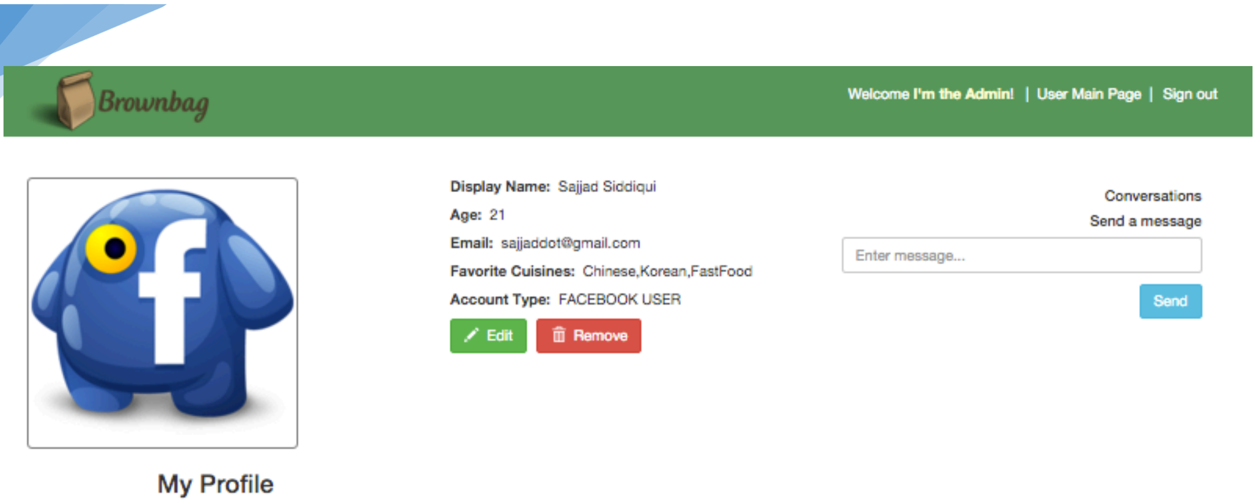
User Feedback

User1 rated 5 stars
I love mcdeez

Jahandar Jamalli rated 4 stars
Not bad :)

Sajjad Siddiqui rated 4.5 stars
review

Jahandar Jamalli rated 4 stars
hgkgkgk



Display Name: Sajjad Siddiqui
Age: 21
Email: sajjaddot@gmail.com
Favorite Cuisines: Chinese,Korean,FastFood
Account Type: FACEBOOK USER

My Profile

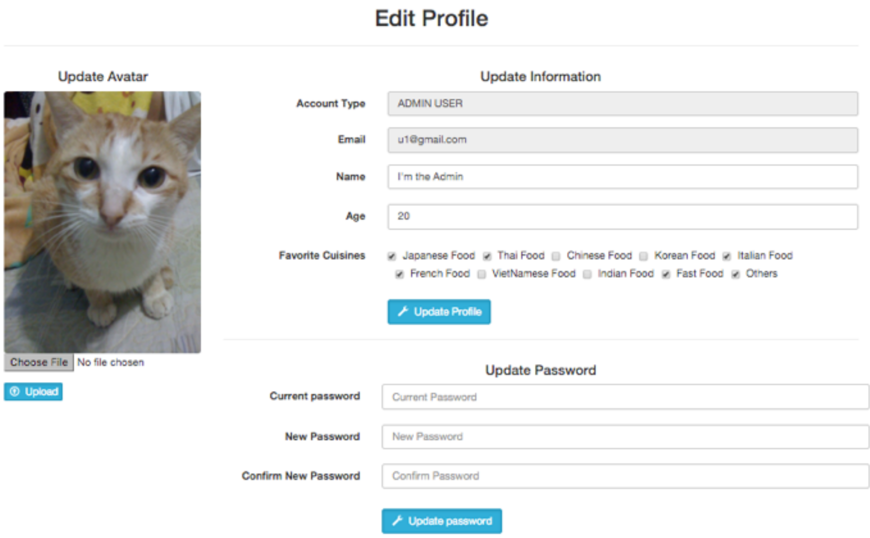
Above is the restaurant profile page from the view of an **admin**, it has the following UI components:

- The user avatar at the top left.
- The user profile information at the middle.
- And the conservation section for sending message.

/users/:id/edit: user profile edit page, Restaurants and regular users have very similar profile editing page.

On the right is the full edit profile, when the admin tries to edit his/her own profile. It has the following UI component:

- The avatar updating section.
- The profile information updating section
- The password updating section.



Edit Profile

Update Avatar

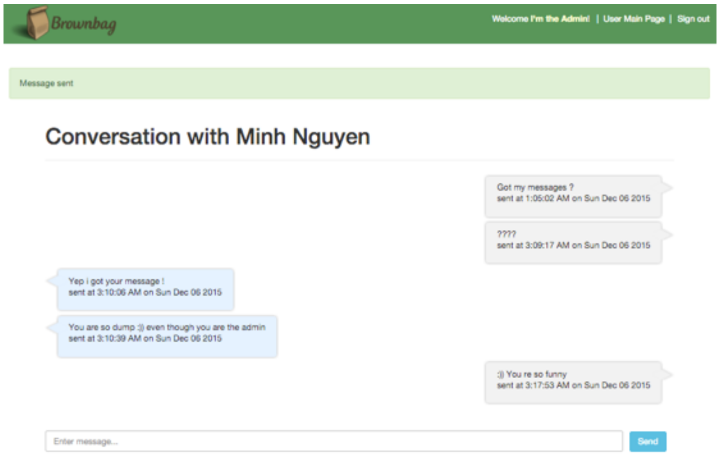
Update Information

Account Type: ADMIN USER
Email: u1@gmail.com
Name: I'm the Admin
Age: 20
Favorite Cuisines: ☑ Japanese Food ☑ Thai Food □ Chinese Food □ Korean Food ☑ Italian Food ☑ French Food □ VietNamese Food □ Indian Food ☑ Fast Food ☑ Others

Update Password

Current password: Current Password
New Password: New Password
Confirm New Password: Confirm Password

/conservation/:email: chat page, where email is the email of recipient.



Message sent

Conversation with Minh Nguyen

Got my messages ?
sent at 1:05:02 AM on Sun Dec 06 2015

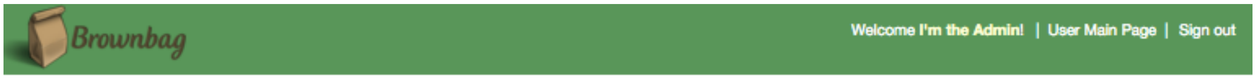
????
sent at 3:09:17 AM on Sun Dec 06 2015

Yep i got your message !
sent at 3:10:06 AM on Sun Dec 06 2015

You are so dump :)) even though you are the admin
sent at 3:10:39 AM on Sun Dec 06 2015

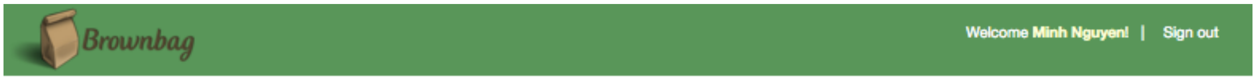
:)) You re so funny
sent at 3:17:53 AM on Sun Dec 06 2015

On the right of the chat box are the messages that you send to someone, and on the left are their replies.



The header of the admin:

- Clicking on the brownbag logo will redirect to /users/admin.
- Clicking on the admin name – "I'm the Admin", will send the admin to the admin profile page.
- Clicking on "User Main Page" will redirect the admin to /users/main.
- Clicking "Sign Out" will sign the admin out of the system back to the login screen.



The header of a user (restaurant, Facebook and regular user account):

- Clicking on the brownbag logo will redirect to /users/main.
- Clicking on the user name "Minh Nguyen" will send the user to the profile page.
- Clicking "Sign Out" will sign the user out of system.

Security Issues Covered:

Security issue 1: SQL injection

Treatment:

We used mongoose which creates BSON objects rather than normal a string. This prevents from a malicious user from performing SQL injection on our databases.

$where permits you to run arbitrary JavaScript expressions directly on the server, but we have not used it. So our database is secure.

Evidence:

Straight from the developers' mouth

https://docs.mongodb.org/manual/faq/developers/ - how-does-mongodb-address-sql-or-query-injection

Security issue 2: Authorization

Treatment:

We control the access of the users to the resource based on the Auth._id saved in their session. So when ever a user makes a request to the server, we use that Auth._id to identify their account type and thus their right to perform an action. For instance, a regular user is prevented from modifying other user profile, a regular is not given access to the admin page or the admin cannot modify someone password.

Evidence:

If you run our **unit test**, you will see there is a section (test case 36 → 40) where we test specifically for this issue.

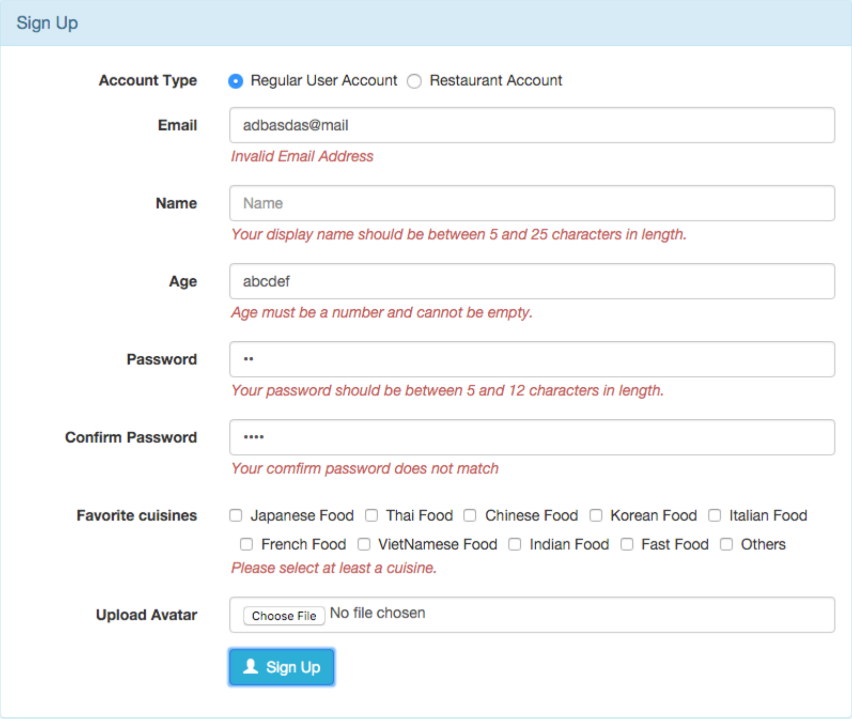
Security issue 3: Authentication

Treatment:

We have variety of input checks when a user signs up for an account in the front end, that prevent the user from inputting invalid format email or short password that is easy to guess. As always, we ask for the user to type in their password twice, to make sure that they don't make any typo in their first try. Duplicated email is also not allowed.

Since it is possible for that the user may directly post a request to the server to create an account without using the interface, we have the same set of input checks in the backend. So there is no way that someone can input malicious input to our database.

Evidence:
I was trying to sign up for an account with malicious input, and here is what I got:



Sign Up

Account Type ● Regular User Account ○ Restaurant Account

Email adbasdas@mail
Invalid Email Address

Name Name
Your display name should be between 5 and 25 characters in length.

Age abcdef
Age must be a number and cannot be empty.

Password ••
Your password should be between 5 and 12 characters in length.

Confirm Password ••••
Your comfirm password does not match

Favorite cuisines □ Japanese Food □ Thai Food □ Chinese Food □ Korean Food □ Italian Food □ French Food □ VietNamese Food □ Indian Food □ Fast Food □ Others
Please select at least a cuisine.

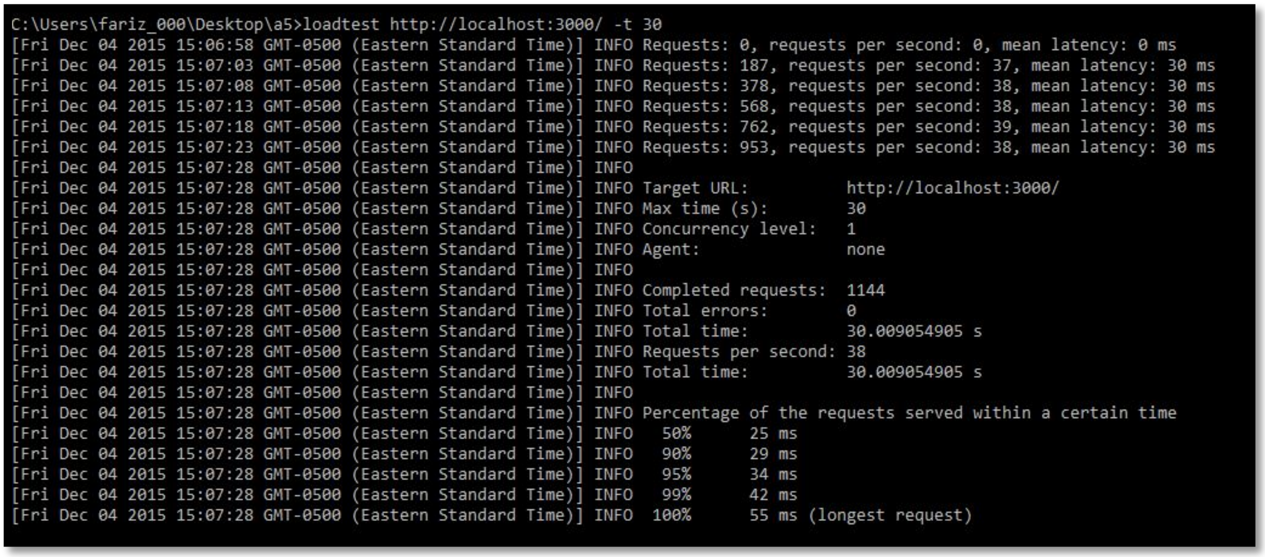
Upload Avatar Choose File No file chosen

Sign Up

Performance Test and Optimization:

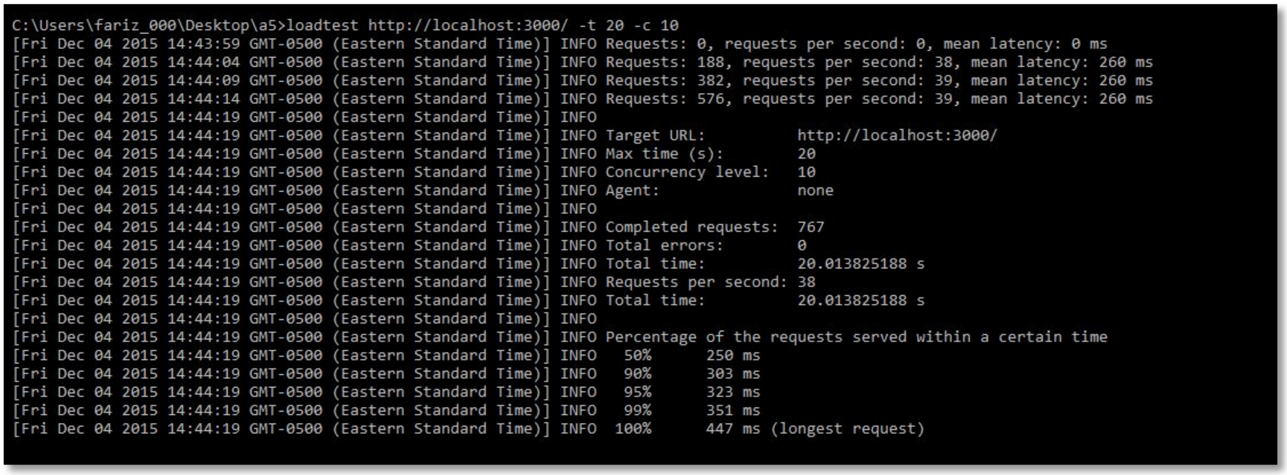
We tested performance of our website (Brown Bag) under different concurrency levels for different amount of time. Below are the stress test results:

Concurrency level 1 for 30 seconds:



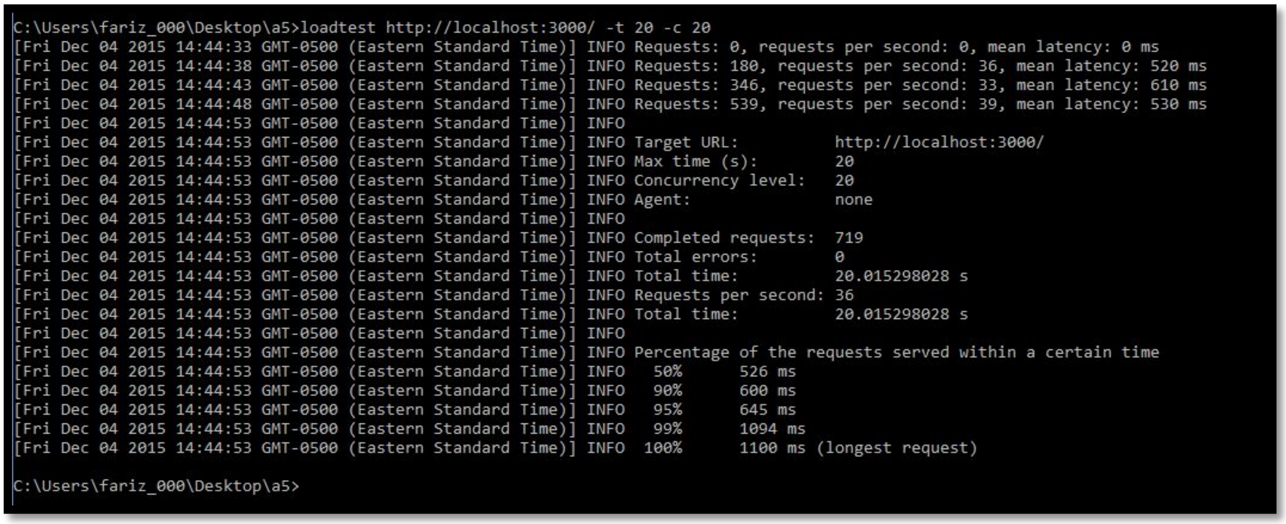
```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 30
[Fri Dec 04 2015 15:06:58 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 15:07:03 GMT-0500 (Eastern Standard Time)] INFO Requests: 187, requests per second: 37, mean latency: 30 ms
[Fri Dec 04 2015 15:07:08 GMT-0500 (Eastern Standard Time)] INFO Requests: 378, requests per second: 38, mean latency: 30 ms
[Fri Dec 04 2015 15:07:13 GMT-0500 (Eastern Standard Time)] INFO Requests: 568, requests per second: 38, mean latency: 30 ms
[Fri Dec 04 2015 15:07:18 GMT-0500 (Eastern Standard Time)] INFO Requests: 762, requests per second: 39, mean latency: 30 ms
[Fri Dec 04 2015 15:07:23 GMT-0500 (Eastern Standard Time)] INFO Requests: 953, requests per second: 38, mean latency: 30 ms
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        30
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   1
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  1144
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Total time:          30.009054905 s
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 38
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Total time:          30.009054905 s
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO   50%      25 ms
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO   90%      29 ms
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO   95%      34 ms
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO   99%      42 ms
[Fri Dec 04 2015 15:07:28 GMT-0500 (Eastern Standard Time)] INFO  100%      55 ms (longest request)
```

Concurrency level 10 for 20 seconds



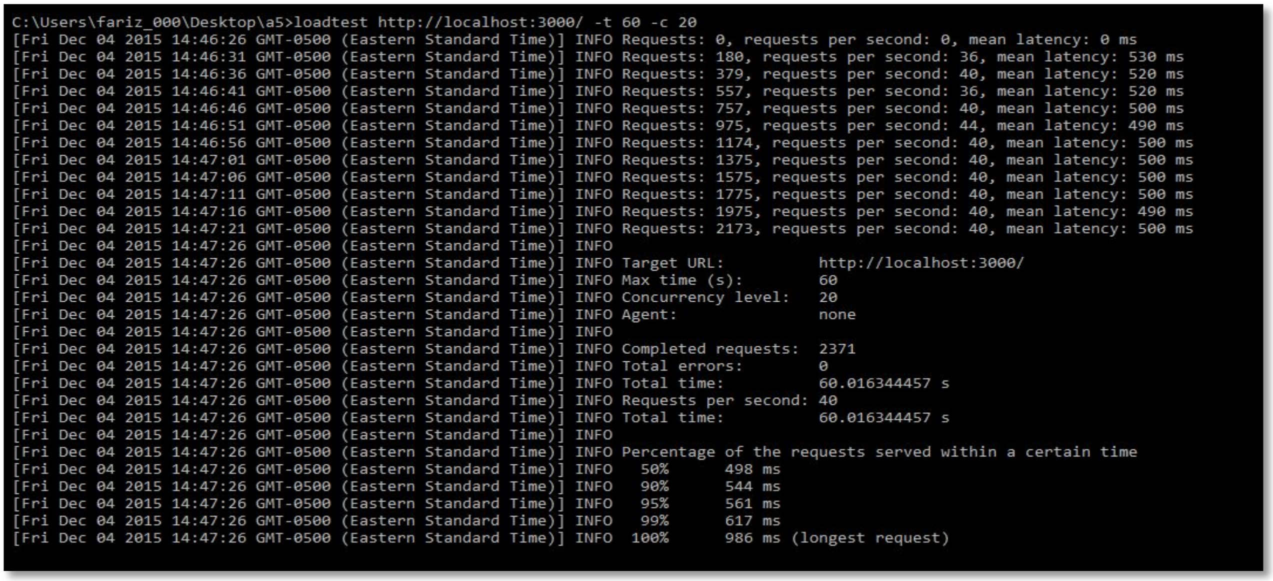
```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 20 -c 10
[Fri Dec 04 2015 14:43:59 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:44:04 GMT-0500 (Eastern Standard Time)] INFO Requests: 188, requests per second: 38, mean latency: 260 ms
[Fri Dec 04 2015 14:44:09 GMT-0500 (Eastern Standard Time)] INFO Requests: 382, requests per second: 39, mean latency: 260 ms
[Fri Dec 04 2015 14:44:14 GMT-0500 (Eastern Standard Time)] INFO Requests: 576, requests per second: 39, mean latency: 260 ms
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        20
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   10
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  767
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.013825188 s
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 38
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.013825188 s
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO   50%      250 ms
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO   90%      303 ms
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO   95%      323 ms
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO   99%      351 ms
[Fri Dec 04 2015 14:44:19 GMT-0500 (Eastern Standard Time)] INFO  100%      447 ms (longest request)
```

Concurrency level 20 for 20 seconds:



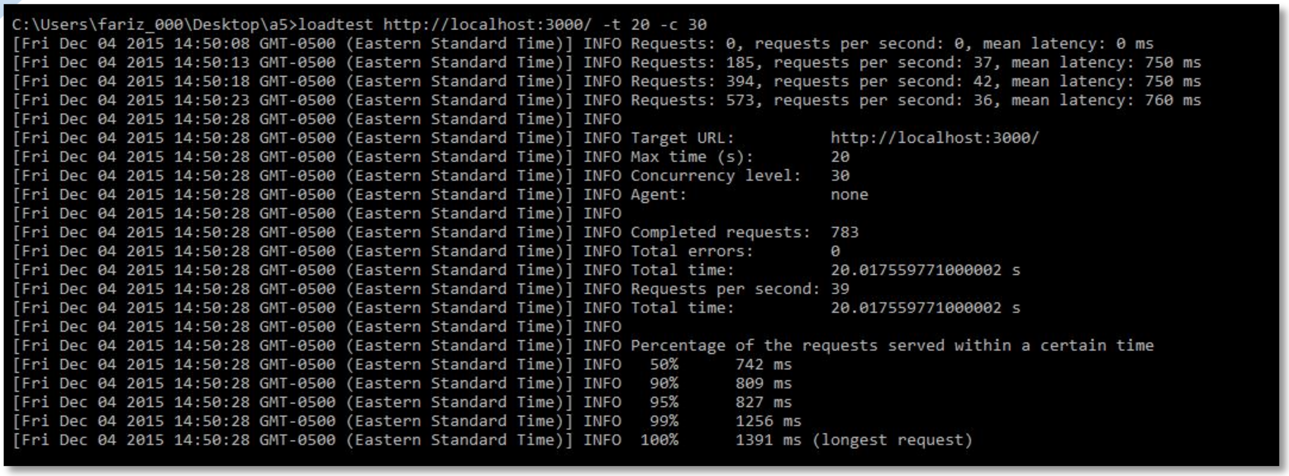
```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 20 -c 20
[Fri Dec 04 2015 14:44:33 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:44:38 GMT-0500 (Eastern Standard Time)] INFO Requests: 180, requests per second: 36, mean latency: 520 ms
[Fri Dec 04 2015 14:44:43 GMT-0500 (Eastern Standard Time)] INFO Requests: 346, requests per second: 33, mean latency: 610 ms
[Fri Dec 04 2015 14:44:48 GMT-0500 (Eastern Standard Time)] INFO Requests: 539, requests per second: 39, mean latency: 530 ms
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        20
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   20
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  719
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.015298028 s
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 36
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.015298028 s
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO   50%      526 ms
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO   90%      600 ms
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO   95%      645 ms
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO   99%      1094 ms
[Fri Dec 04 2015 14:44:53 GMT-0500 (Eastern Standard Time)] INFO  100%      1100 ms (longest request)

C:\Users\fariz_000\Desktop\a5>
```

Concurrency level 20 for 60 seconds



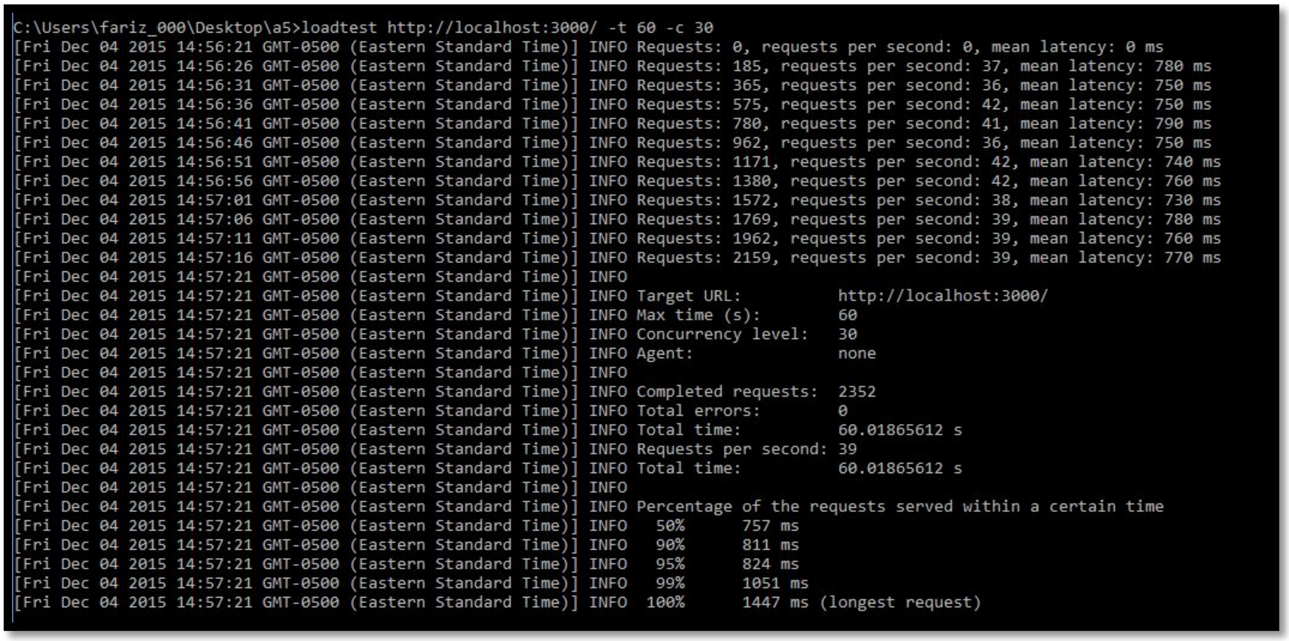
```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 60 -c 20
[Fri Dec 04 2015 14:46:26 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:46:31 GMT-0500 (Eastern Standard Time)] INFO Requests: 180, requests per second: 36, mean latency: 530 ms
[Fri Dec 04 2015 14:46:36 GMT-0500 (Eastern Standard Time)] INFO Requests: 379, requests per second: 40, mean latency: 520 ms
[Fri Dec 04 2015 14:46:41 GMT-0500 (Eastern Standard Time)] INFO Requests: 557, requests per second: 36, mean latency: 520 ms
[Fri Dec 04 2015 14:46:46 GMT-0500 (Eastern Standard Time)] INFO Requests: 757, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:46:51 GMT-0500 (Eastern Standard Time)] INFO Requests: 975, requests per second: 44, mean latency: 490 ms
[Fri Dec 04 2015 14:46:56 GMT-0500 (Eastern Standard Time)] INFO Requests: 1174, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:47:01 GMT-0500 (Eastern Standard Time)] INFO Requests: 1375, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:47:06 GMT-0500 (Eastern Standard Time)] INFO Requests: 1575, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:47:11 GMT-0500 (Eastern Standard Time)] INFO Requests: 1775, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:47:16 GMT-0500 (Eastern Standard Time)] INFO Requests: 1975, requests per second: 40, mean latency: 490 ms
[Fri Dec 04 2015 14:47:21 GMT-0500 (Eastern Standard Time)] INFO Requests: 2173, requests per second: 40, mean latency: 500 ms
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        60
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   20
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  2371
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.016344457 s
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 40
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.016344457 s
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO   50%      498 ms
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO   90%      544 ms
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO   95%      561 ms
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO   99%      617 ms
[Fri Dec 04 2015 14:47:26 GMT-0500 (Eastern Standard Time)] INFO  100%      986 ms (longest request)
```

Concurrency level 30 for 20 seconds



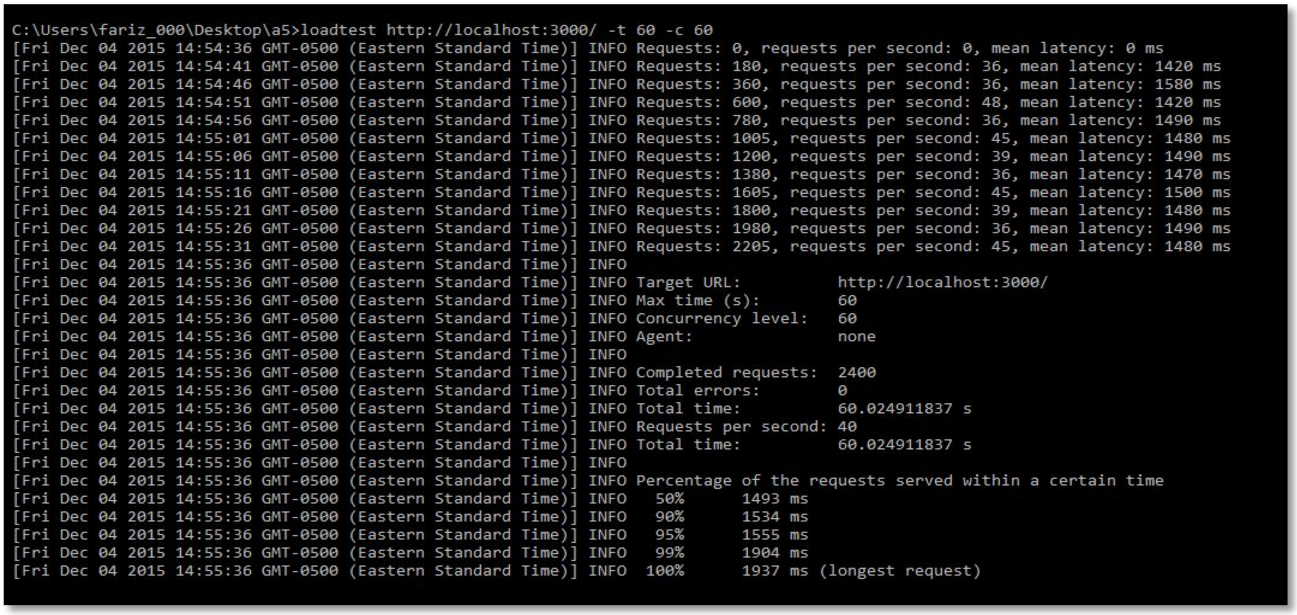
```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 20 -c 30
[Fri Dec 04 2015 14:50:08 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:50:13 GMT-0500 (Eastern Standard Time)] INFO Requests: 185, requests per second: 37, mean latency: 750 ms
[Fri Dec 04 2015 14:50:18 GMT-0500 (Eastern Standard Time)] INFO Requests: 394, requests per second: 42, mean latency: 750 ms
[Fri Dec 04 2015 14:50:23 GMT-0500 (Eastern Standard Time)] INFO Requests: 573, requests per second: 36, mean latency: 760 ms
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        20
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   30
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  783
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.017559771000002 s
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 39
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Total time:          20.017559771000002 s
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO   50%      742 ms
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO   90%      809 ms
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO   95%      827 ms
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO   99%      1256 ms
[Fri Dec 04 2015 14:50:28 GMT-0500 (Eastern Standard Time)] INFO  100%      1391 ms (longest request)
```

Concurrency level 30 for 60 seconds



```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 60 -c 30
[Fri Dec 04 2015 14:56:21 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:56:26 GMT-0500 (Eastern Standard Time)] INFO Requests: 185, requests per second: 37, mean latency: 780 ms
[Fri Dec 04 2015 14:56:31 GMT-0500 (Eastern Standard Time)] INFO Requests: 365, requests per second: 36, mean latency: 750 ms
[Fri Dec 04 2015 14:56:36 GMT-0500 (Eastern Standard Time)] INFO Requests: 575, requests per second: 42, mean latency: 750 ms
[Fri Dec 04 2015 14:56:41 GMT-0500 (Eastern Standard Time)] INFO Requests: 780, requests per second: 41, mean latency: 790 ms
[Fri Dec 04 2015 14:56:46 GMT-0500 (Eastern Standard Time)] INFO Requests: 962, requests per second: 36, mean latency: 750 ms
[Fri Dec 04 2015 14:56:51 GMT-0500 (Eastern Standard Time)] INFO Requests: 1171, requests per second: 42, mean latency: 740 ms
[Fri Dec 04 2015 14:56:56 GMT-0500 (Eastern Standard Time)] INFO Requests: 1380, requests per second: 42, mean latency: 760 ms
[Fri Dec 04 2015 14:57:01 GMT-0500 (Eastern Standard Time)] INFO Requests: 1572, requests per second: 38, mean latency: 730 ms
[Fri Dec 04 2015 14:57:06 GMT-0500 (Eastern Standard Time)] INFO Requests: 1769, requests per second: 39, mean latency: 780 ms
[Fri Dec 04 2015 14:57:11 GMT-0500 (Eastern Standard Time)] INFO Requests: 1962, requests per second: 39, mean latency: 760 ms
[Fri Dec 04 2015 14:57:16 GMT-0500 (Eastern Standard Time)] INFO Requests: 2159, requests per second: 39, mean latency: 770 ms
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        60
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   30
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  2352
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.01865612 s
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 39
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.01865612 s
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO   50%      757 ms
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO   90%      811 ms
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO   95%      824 ms
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO   99%      1051 ms
[Fri Dec 04 2015 14:57:21 GMT-0500 (Eastern Standard Time)] INFO  100%      1447 ms (longest request)
```

Concurrency level 60 for 60 seconds



```
C:\Users\fariz_000\Desktop\a5>loadtest http://localhost:3000/ -t 60 -c 60
[Fri Dec 04 2015 14:54:36 GMT-0500 (Eastern Standard Time)] INFO Requests: 0, requests per second: 0, mean latency: 0 ms
[Fri Dec 04 2015 14:54:41 GMT-0500 (Eastern Standard Time)] INFO Requests: 180, requests per second: 36, mean latency: 1420 ms
[Fri Dec 04 2015 14:54:46 GMT-0500 (Eastern Standard Time)] INFO Requests: 360, requests per second: 36, mean latency: 1580 ms
[Fri Dec 04 2015 14:54:51 GMT-0500 (Eastern Standard Time)] INFO Requests: 600, requests per second: 48, mean latency: 1420 ms
[Fri Dec 04 2015 14:54:56 GMT-0500 (Eastern Standard Time)] INFO Requests: 780, requests per second: 36, mean latency: 1490 ms
[Fri Dec 04 2015 14:55:01 GMT-0500 (Eastern Standard Time)] INFO Requests: 1005, requests per second: 45, mean latency: 1480 ms
[Fri Dec 04 2015 14:55:06 GMT-0500 (Eastern Standard Time)] INFO Requests: 1200, requests per second: 39, mean latency: 1490 ms
[Fri Dec 04 2015 14:55:11 GMT-0500 (Eastern Standard Time)] INFO Requests: 1380, requests per second: 36, mean latency: 1470 ms
[Fri Dec 04 2015 14:55:16 GMT-0500 (Eastern Standard Time)] INFO Requests: 1605, requests per second: 45, mean latency: 1500 ms
[Fri Dec 04 2015 14:55:21 GMT-0500 (Eastern Standard Time)] INFO Requests: 1800, requests per second: 39, mean latency: 1480 ms
[Fri Dec 04 2015 14:55:26 GMT-0500 (Eastern Standard Time)] INFO Requests: 1980, requests per second: 36, mean latency: 1490 ms
[Fri Dec 04 2015 14:55:31 GMT-0500 (Eastern Standard Time)] INFO Requests: 2205, requests per second: 45, mean latency: 1480 ms
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Target URL:          http://localhost:3000/
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Max time (s):        60
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Concurrency level:   60
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Agent:               none
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Completed requests:  2400
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Total errors:        0
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.024911837 s
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Requests per second: 40
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Total time:          60.024911837 s
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO Percentage of the requests served within a certain time
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO   50%      1493 ms
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO   90%      1534 ms
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO   95%      1555 ms
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO   99%      1904 ms
[Fri Dec 04 2015 14:55:36 GMT-0500 (Eastern Standard Time)] INFO  100%      1937 ms (longest request)
```

According to the results from stress testing we can see that under same level of stress/concurrency our website serves requests with almost the same performance even if we increase the time. However, when we increase the load, it takes more time for our web application to respond request which is natural.

Note: *loadtest npm* module has been used for performance testing.

Optimization techniques used:

- We store user data in several different tables (e.g. Auth, User, FBUser, Restaurant, Review…), so that when we need to run query on the data, we don't have go through data of every user in the system. We just go straight to the table that we need to find the data from, and extract the data from there. It is done by maintaining the common foreign key Auth._id across the tables.
- We compressed the images size, for instance, our logo image (brown bag), used to weight 25 mega bytes, was reduced to just half a mega byte.
- We **minified** our script and style sheet files that get sent to the user, so all the codes are now in one long line. This reduce the file size that gets sent to the user, thus a gain in performance. We included the non-minified style sheet and java script files, in the same folder, in case you want to look at them in a more readable format.

We published our app on heroku: http://a5brownbag.herokuapp.com/ . If you want to do some performance test, we recommend you to go there because it already has an existing database with about 10 users in it.

Video Demo Link (YouTube):

https://www.youtube.com/watch?v=aa7e3b8zSBM

Instruction to run unit test

- "npm install" to install all dependencies.
- Run your mongodb database.
- Open a new shell, run the web app by the the the command "node server.js".
- Open another shell, run the unit test by the command "npm test".
- Wait approximately 5 to 10 seconds until the test result shows up.
- There 40 test cases in total, our app passed all of them.

Resource:

Facebook user logo:
http://sergeybuslaev.ru/wp-content/uploads/2014/02/facebook.png

Tool for compressing files:
http://www.giftofspeed.com/javascript-compressor/