

Title:

InstaFood

Description of topic:

I came up with the idea of creating a platform to connect foodies who are tired of spending on restaurants with home chefs who could gain monetarily for their efforts.

InstaFood will let a cook snap a picture of the meal he cooked and upload it. A hungry person can just open the app, browse through the pictures put up by various cooks, and order his home-made meal.

Website functionalities:

Every user can play either of/or both the roles:

1. Buyer (user who wants to buy food)
2. Seller (user who wants to sell food)

Roles for Buyer:

- Register on the website- user name, first name, last name, gender, email ID.
- Sign into the site
- Browse through various food items from multiple cuisines
- Select food item(s) and view Order summary
- Confirm payment
(On confirmation an email will be sent with details or the location and time for delivery)
- Can view order history

Roles for Seller:

- Register on the website- - user name, first name, last name, gender, email ID.
- Sign into the site
- Upload a food item by mentioning the appropriate food type, cost, time for delivery
- On receiving a notification- accept the order
- Confirm delivery of the item
- Fulfil pending orders
- Can view posted items list

Technical Details:

We developed our website using MEAN STACK (MongoDB, express js, angular js and node js) on windows platform.

➤ Installations:

- Installed and used text editors – IntelliJ and sublime text
- MongoDB installation
- Git bash installation for running mongodb commands
- Nodejs installation
- Node packages installations (these can be installed through command line statements).

The main reason we chose MEAN:

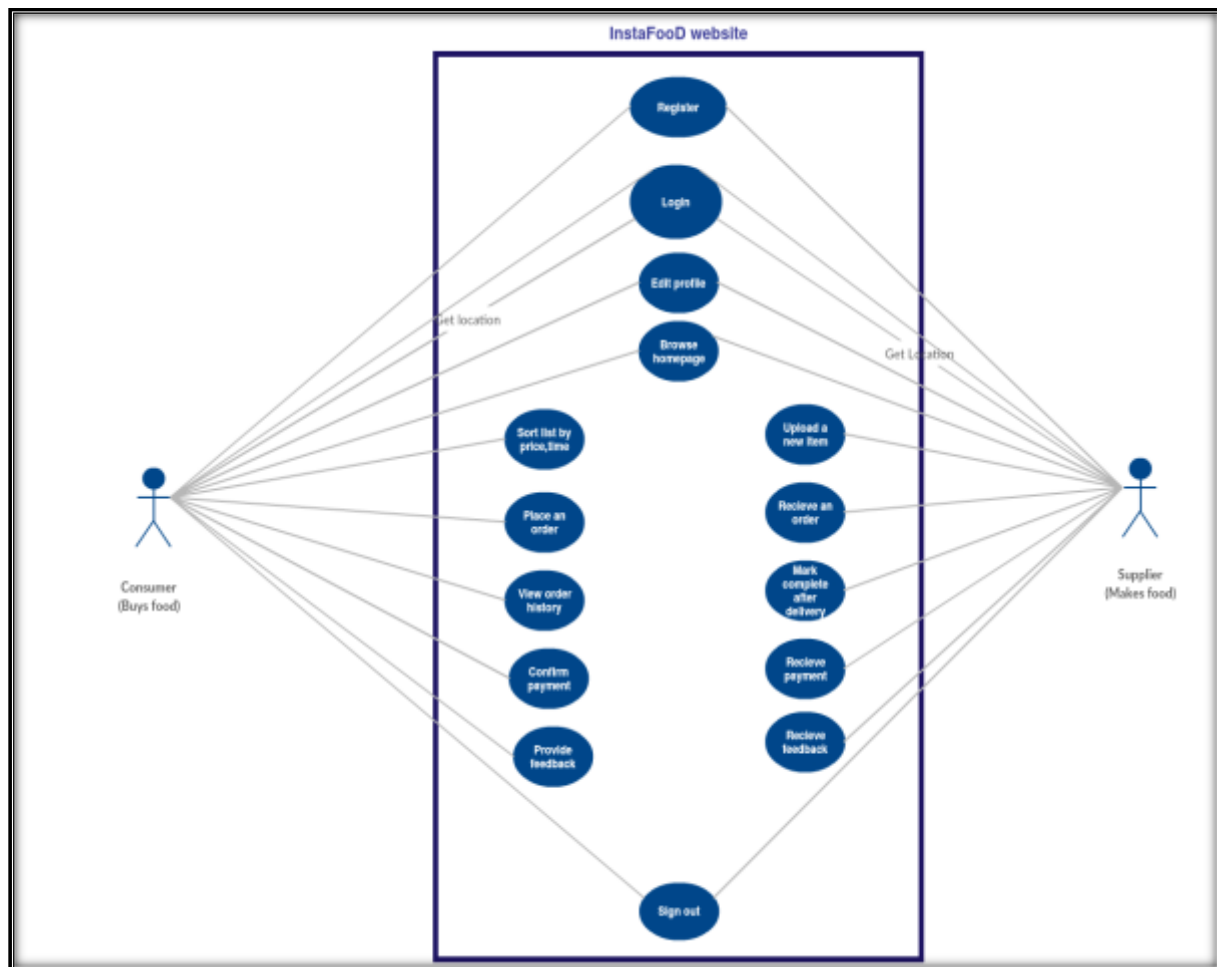
- Open source and can be used in any operating system.
- Supports MVC (model view controller architecture)
- MongoDB is a schemaless database framework which doesn't involve any complex joins and is just one group holding different collections.
- Each object ID created in the database is encrypted.
- Node.js platform provides a wide range of features that can be carried out (with simple command line installations of node modules).
- Multiple-platform compatible written code
- Javascript is used everywhere, making it more convenient - client side as well as server side and mongoDB uses JSON format

Key features:

- User registration, login and logout.
- Basic implementation of StripeJS for payment transaction functionality. It sends the customer's payment information from the client to Stripe's server and returns a token that Stripe associated with that payment information. Then, when the form is submitted, the token can be used on our site to process the payment. The customer's payment information never reaches our server thereby protecting the customer's information.
- Implemented responsive web design using Bootstrap
- Using PassportJS as authentication middleware to take care of our authentication concerns as it is easily configured into any Express based web application
- Users can edit their profile (change password, address)
- Search for food based on food variety
- Database collections part of the code implemented in models
- Allowing seller to upload image and re-size it accordingly to 600 X 400
- Email notification to the buyer and seller once an order has been requested using nodemailer.
- Allow the seller to view the previously added postings.

- Allow the users to view the order history.
- Options for the users to search food.
- Dialogs for every required field (register, sign in, placing order etc.)
- Intellisense for the users while entering address using address parser.
- Option for the seller to confirm once the food is checked out.
- Handled cases such as seller ordering his own food by mistake
- Prevents seller from editing food that has already been ordered.
- Alerts through dialogs when invalid credentials are entered or incorrect password is given.
- Displays confirmation messages after successful orders/payments/uploads.
- Dialogs that prompt users for searching items/posting images/making payments.
- Immediate update of available food list by removing items that have been ordered by others.

Various Modules available to the users:



References used for installations:

- MongoDB :
 - <https://www.mongodb.org/downloads>
 - <https://docs.mongodb.org/manual/>
 - <https://git-scm.com/downloads>
- Node.js and packages:
 - <http://www.bradoncode.com/tutorials/mean-stack-tutorial-part-1-setup/>
 - <https://nodejs.org/en/>
 - Node packages and bower components – expressjs, stripejs, mongoose, passportjs, node mailer, body parser, address parser,

Note:

I only worked on the project folder. The rest of the folders- node modules and bower components were automatically generated after installing the packages.

I referred through the below websites and videos and created this application:

References:

- <https://www.youtube.com/watch?v=jtaSRzP0i30>
- <https://scotch.io/tutorials/easy-node-authentication-google>
- <http://stackoverflow.com/questions/24352975/passport-google-oauth-on-localhost>
- <http://www.bradoncode.com/tutorials/learn-mean-stack-tutorial/>
- <http://jasonwatmore.com/post/2015/12/09/MEAN-Stack-User-Registration-and-Login-Example.aspx>
- http://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
- <http://stackoverflow.com/questions/30682461/how-do-you-send-emails-from-angular-controllers-on-the-mean-js-stack>
- <https://stripe.com/docs/custom-form>
- Super MEAN Stack Tutorial: Angular, Node/Express, Webpack, MongoDB, SASS, Babel/ES6, Bootstrap- <https://www.youtube.com/watch?v=6Sbau-oE37w>
- Introduction to Passport JS: <https://www.youtube.com/watch?v=jtaSRzP0i30>
- <http://stackoverflow.com/questions/22579041/mean-stack-pass-form-from-angular-to-express-to-be-emailed-by-nodemailer>
- Tutorial for Social Network app using MEAN:
 - <https://www.youtube.com/watch?v=lcHlywuHNwI>
 - <https://www.youtube.com/watch?v=aM18UsOFZw8>
 - <https://www.youtube.com/watch?v=USKaHwICQX4>
 - <https://www.youtube.com/watch?v=WySkMV7Vz-8>
- <https://stripe.com/docs/examples>

- [File uploads: http://stackoverflow.com/questions/22941535/mean-stack-file-uploads](http://stackoverflow.com/questions/22941535/mean-stack-file-uploads)
- <https://stripe.com/docs/stripe.js/switching>
- <http://stackoverflow.com/questions/2142535/how-to-clear-the-canvas-for-redrawing>
- <https://thecodebarbarian.wordpress.com/2013/07/22/introduction-to-the-mean-stack-part-one-setting-up-your-tools/>
- <https://thecodebarbarian.wordpress.com/2013/07/29/introduction-to-the-mean-stack-part-two-building-and-testing-a-to-do-list/>
- <https://thecodebarbarian.wordpress.com/2013/04/29/easy-web-prototyping-with-mongodb-and-nodejs/>
- <https://thinkster.io/mean-stack-tutorial>
- <https://devcenter.heroku.com/articles/mean-apps-restful-api>
- <https://scotch.io/tutorials/setting-up-a-mean-stack-single-page-application>
- <http://meanjs.org/>
- <https://www.youtube.com/watch?v=OhPFgqHz68o>
- [MEAN Stack Tutorial: Getting started for beginners - https://www.youtube.com/watch?v=rIVwJvE3C5A](https://www.youtube.com/watch?v=rIVwJvE3C5A)
- [Building a real world application from scratchhttps://www.youtube.com/watch?v=8CK93h5YJp0](https://www.youtube.com/watch?v=8CK93h5YJp0)
- [MEAN Stack Intro: Build an end-to-end application https://www.youtube.com/watch?v=AEE7DY2AYvI](https://www.youtube.com/watch?v=AEE7DY2AYvI)
- [How to build a M.E.A.N web application-https://www.youtube.com/watch?v=OhPFgqHz68o](https://www.youtube.com/watch?v=OhPFgqHz68o)