



**By: Tomi, Moon Hwan, Ermias and Suhye**

# GREEN THUMBS AND CLIENT SPECIFICATIONS

Green thumbs is an open source project in creating a **WEB** interface. It rates food-vending organizations on sustainability and eco-friendliness.

## Project Specifications

### Client Requirements:

- A login/logout feature.
- Allow users to upload an organization of their choosing.
- Allow users to rate organizations using thumbs.
- Allow for open collaboration - open source it on Github.
- Have a simple page on climate change/facts.

Client has no plans for publication anytime soon - wanted a basic outline

---

# DEMO

**Green Thumbs**

**CLIMATE CHANGE AWARENESS**

**REVIEWS**

**LOGIN**

**SIGN UP**

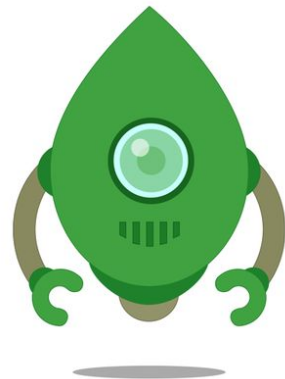


## Welcome!

Here at **GreenThumbs**, we are on a mission to make food consumption sustainable. With our **green thumb** review system, you can measure just how sustainable your food consumption is and make **~greener choices~**

Join our vibrant community of **green** eaters and **green** reviewers today!

# BACKEND CHOICES



Templating: EJS

Middleware/Libraries: Passport, Express (including express-session), Cookie-parser

Database: MongoDB (Mongoose)

Other Helpful Technologies: Facebook App (Developer), RoboMongo

# USING PASSPORT/USER AUTHENTICATION

## Client Demands

- A safe and easy login feature
- Allow some way of verifying users.
- NOT anonymous – Open dialogue

## Our Take

- Used Facebook App.
- Authenticate users using passport.
- Used Flash.JS to redirect users if they come across errors.

## Issues and Future Plans:

- Limited to 50 users on a free account.
- Had to literally send in a picture of a passport – issue with handing over code.
- Plan to add google and more ways of authentication using other authentication system.
- Do people trust Facebook enough? – User assumptions

# DATABASE AND STORAGE

## Client Demands

- Way of reading reviews
- Way of reading profiles of organizations
- Way of logging in

## Our Take

- Used Mongoose as we could easily define schemas to what we need.
- Different schemas for users and organizations.

```
var mongoose = require('mongoose');

var organizationSchema = mongoose.Schema({
  name: { type: String },
  address: { type: String },
  city: { type: String },
  country: { type: String },
  sector: { type: String },
  website: { type: String },
  image: { type: String, default: 'grey.jpeg' },
  employees: [{
    employeeId: { type: String, default: '' },
    employeeFullname: { type: String, default: '' },
    employeeRole: { type: String, default: '' }
  }],

  organizationRating: [{
    organizationName: { type: String, default: '' },
    userFullname: { type: String, default: '' },
    userRole: { type: String, default: '' },
    organizationImage: { type: String, default: '' },
    userRating: { type: Number, default: 0 },
    userReview: { type: String, default: '' }
  }],

  ratingNumber: [Number],
  ratingSum: { type: Number, default: 0 }
});

module.exports = mongoose.model('Organization', organizationSchema);
```

# MORE DATABASE

- Used robomongo to check storage of data.
- Chose to remove use of pictures at the moment (privacy issues)

The screenshot shows the Studio 3T for MongoDB interface. The top toolbar includes icons for Connect, Collection, IntelliShell, SQL, Aggregate, Map-Reduce, Compare, Schema, Reschema, Tasks, Export, Import, Data Masking, and SQL Migration. A green banner at the top states: "Welcome to your trial of full free access to Studio 3T. The trial is available for 10 days. [How to get a license?](#)".

The left sidebar shows a tree view of the database structure. Under "test localhost:27017 [direct]", there is a folder "Green-Thumbs" containing "Collections (4)": "MyCollection", "organizations", "sessions", and "users". The "organizations" collection is selected.

The main panel displays the "organizations" collection. The "Query" field is empty, and the "Projection" field is set to "{}". The "Sort" field is also empty. The "Skip" field is empty, and the "Limit" field is set to "50". The "Result" tab is active, showing a table view of the documents.

_id	image	ratingNumber	ratingSum	employees	organizationRating	name
606935c931...		[ 0 elements ]	0	[ 0 elements ]	[ 0 elements ]	Test
606b2786e7...		[ 3 elements ]	14	[ 0 elements ]	[ 3 elements ]	ew
606c6d0b85...	Green Thumb...	[ 0 elements ]	0	[ 0 elements ]	[ 0 elements ]	Hi
606d11f3eeb...		[ 0 elements ]	0	[ 0 elements ]	[ 0 elements ]	Test
606e63ef834...	Screen Shot 2...	[ 1 elements ]	5	[ 0 elements ]	[ 1 elements ]	Hi

The bottom status bar indicates "1 document selected" and "Count Documents". The timer shows "00:00:00.015".

# DETAILS ON ORGANIZATION REGISTRATION PAGE

```
module.exports = (app) => {
  app.get('/organization/create', (req, res) => {
    var success = req.flash('success');
    res.render('organization/organization', { title: 'Organization Registration', user: req.user,
    success: success, noErrors: success.length > 0 });
  });

  app.post('/organization/create', (req, res) => {
    var org = new Organization();
    org.name = req.body.name;
    org.address = req.body.address;
    org.city = req.body.city;
    org.country = req.body.country;
    org.sector = req.body.sector;
    org.website = req.body.website;
    org.image = req.body.upload;

    org.save((err) => {
      if (err) {
        console.log(err);
      }

      console.log(org);

      req.flash('success', 'Organization has been successfully registered. ');
      res.redirect('/organization/create')
    })
  });

  app.post('/upload', (req, res) => {
    var form = new formidable.IncomingForm();
    form.uploadDir = path.join(__dirname, '../public/uploads');
    form.on('file', (field, file) => {
      fs.rename(file.path, path.join(form.uploadDir, file.name), (err) => {
        if (err) {
          throw err;
        }
      })
    })
  })
}
```

- Inputs on organization registration form stored in Organization table
- All input fields are required
- Image upload implemented using formidable
- Progress bar for image upload



# DETAILS ON ORGANIZATIONS PAGE / ORGANIZATION PROFILE PAGE

- Average number of thumbs calculated using underscore's reduce function on the array storing all ratings

```
var _ = require('underscore');
var avg;

var arrayAverage = (array) => {
  if (array.length === 0) {
    avg = 0;
  } else {
    avg = _.reduce(array, function (first_num, second_num) {
      return first_num + second_num;
    }, 0) / array.length;
  }

  return avg.toFixed(1);
}
```

# DETAILS ON REVIEW PAGE

```
$(document).ready(function () {  
    var clickedValue = 0;  
  
    // change thumb to filled upon hover  
    $('#thumb1').hover(function () {  
        $('#showTitle').html('Terrible');  
  
        $('#thumb1').attr('src', '/images/filled.png');  
        $('#thumb2').attr('src', '/images/empty.png');  
        $('#thumb3').attr('src', '/images/empty.png');  
        $('#thumb4').attr('src', '/images/empty.png');  
        $('#thumb5').attr('src', '/images/empty.png');  
    });  
  
    // event handler for when a thumb is actually clicked  
    $('#thumb1').on('click', () => {  
        clickedValue = 1;  
    })  
  
    // change thumb to filled upon hover
```

- Event handlers on the five empty thumb images
  - Hover: Image changed to the filled thumb image. Adjective (e.g. 'Average', 'Excellent') displayed on the side.
  - Click: Thumb number (rating) set
- Organization table updated in database with review/rating/reviewer information

# FUTURE DIRECTION

- Implement a location API for when this project gets bigger.
- Two-step authentication process
  - Check authenticity/existence of certain organizations.
- Modify schema for more information.
- Can only search by name, plan to search by filtering.
- Add a rubric for a more robust rating system (we didn't get one from the client but hope to receive it before final handin).
- Allow organizations to comment on user feedback.
- Allow editing/deleting of reviews.

THANK YOU!

QUESTIONS?