

JoinME!

JoinMe! Team

Emma P (lead)

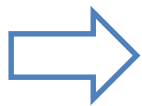
Thomas P

Justin R

Chase L

Paul H

Agenda



- Our motivation/summary
- Application's concept
- Technologies you used
- Design process
- DEMO!
- Future development

MOTIVATION!

Exactly!

MOTIVATION:

- 1. Biggest problem is starting a future goal*
- 2. Biggest problem is progressing to a goal*
- 3. Biggest problem is completing a goal*

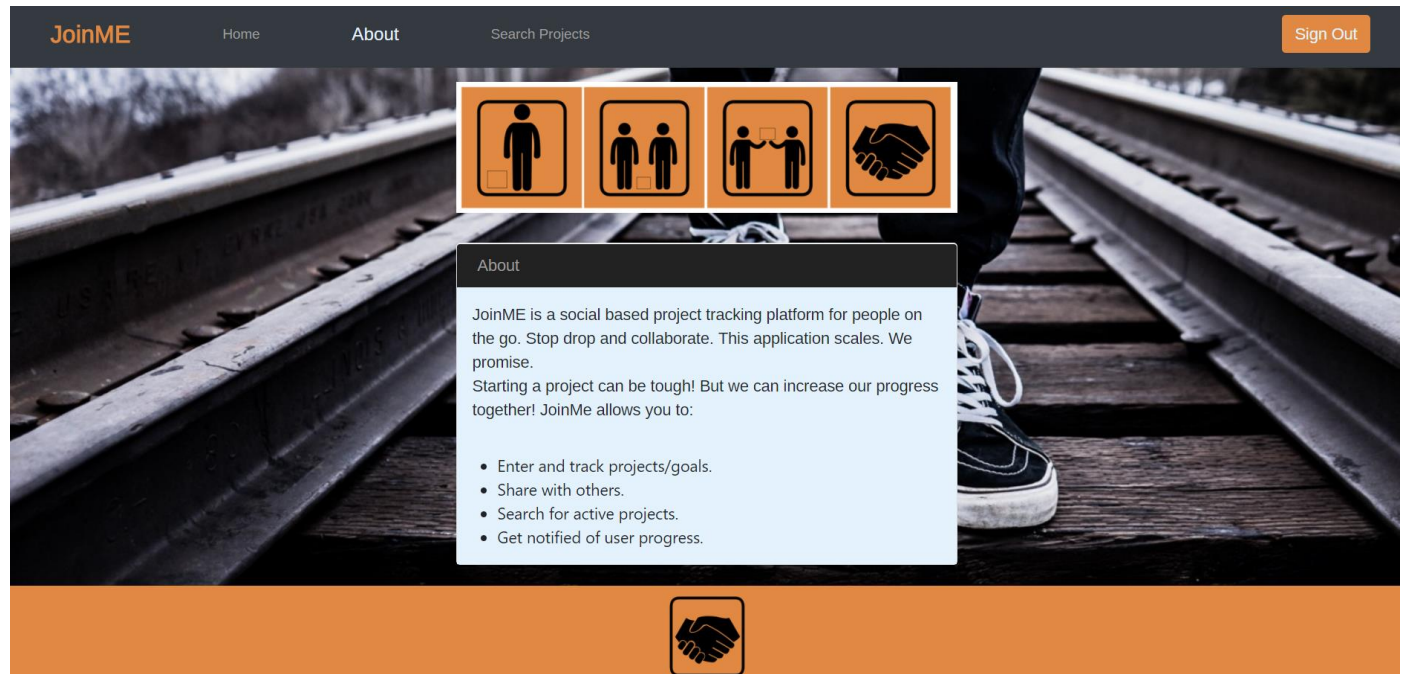
Involving other people provides the motivation!

Concept

User Inputs Project/Goals(steps) !

Application provides:

- 1. Interactive GUI*
- 2. Repository/Tracking*



High Level Coding Summary

Models:

- Database models

Routes:

- API routes for GET,POST,DELETE,PUT
- HTML routes

Styling:

- Used HandleBars and Bootstrap
- Used css for inner styling of specific elements

Logic:

- INDEX: provided checks and routes logic
- SERVER: provided checks and routes logic
- Sequelize: provided some logic

Coding Summary continued NODE packages

Node and Express Web Server:

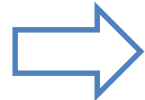
- "bcrypt-nodejs": "0.0.3"
- "body-parser": "^1.18.2",
- "connect-flash": "^0.1.1"
- "dotenv": "^4.0.0"
- "ejs": "^2.5.7"
- "express": "^4.16.2"
- "express-handlebars": "^3.0.0"
- "express-session": "^1.15.6",
- "nodemon": "^1.14.7"
- "passport": "^0.4.0"
- "passport-facebook": "^2.1.1"
- "passport-local": "^1.0.0",
- "path": "^0.12.7"

A LOT of Packages!

MySQL Database with a Sequelize ORM

- "mysql": "^2.15.0",
- "mysql2": "^1.5.1",
- "sequelize": "^4.31.1"

Agenda

- Our motivation/summary
-  • Application's concept
- Technologies you used
- Design process
- DEMO!
- Future development

Concept (Database & Notification)

User Inputs Project/Goal!

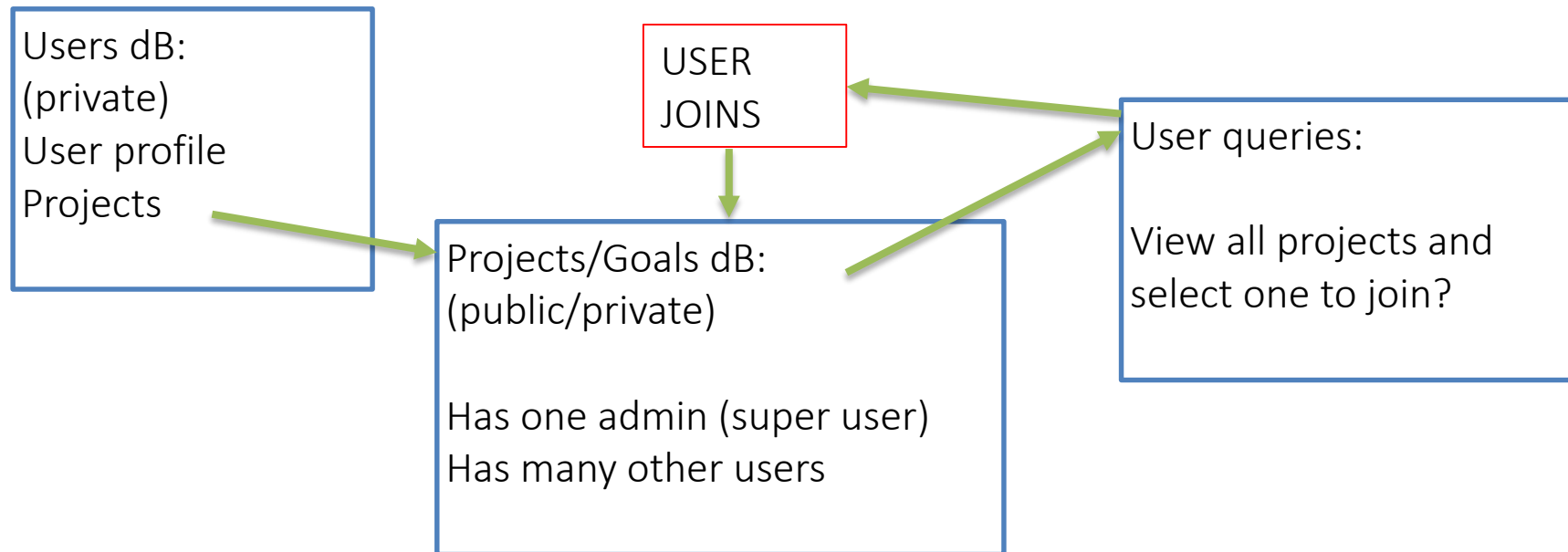
Application creates a record of Goal (steps)

Application allows user to share and track

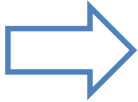
Database activities

1. *User can update their progress*

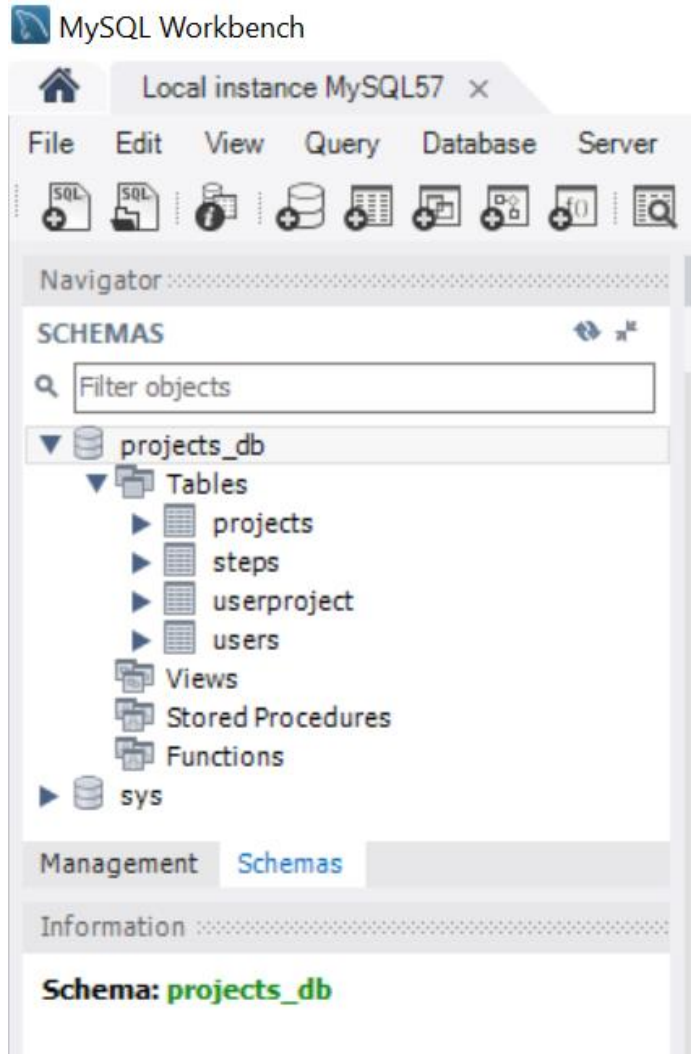
Notification activity



Agenda

- Our motivation/summary
- Application's concept
-  • Technologies you used
- Design process
- DEMO!
- Future development

Projects_db database schema



User table:

- *user information*
- *user belong to projects*

Projects table:

- *Project description*
- *Projects belong to users*

Steps table:

- *Steps description*
- *Completed: Trigger!*
- *Steps belong to users*

Projects_db database relationships & validation

Users:
Defines projects
Has many projects

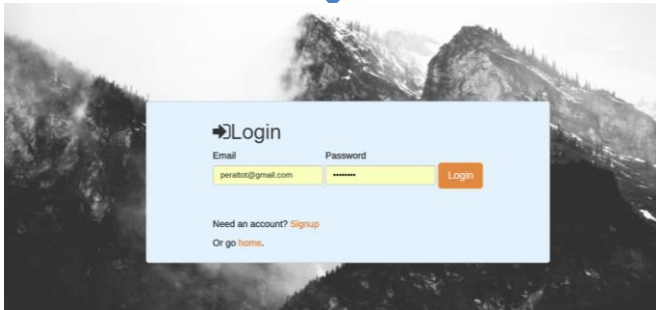
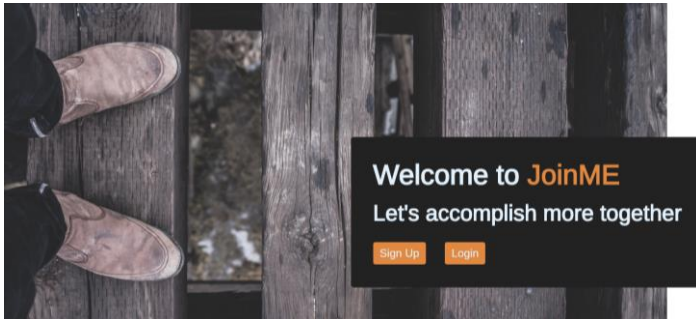
Projects:
Created by user
Can have many users

User_Projects:
Many users can
belong to many
projects

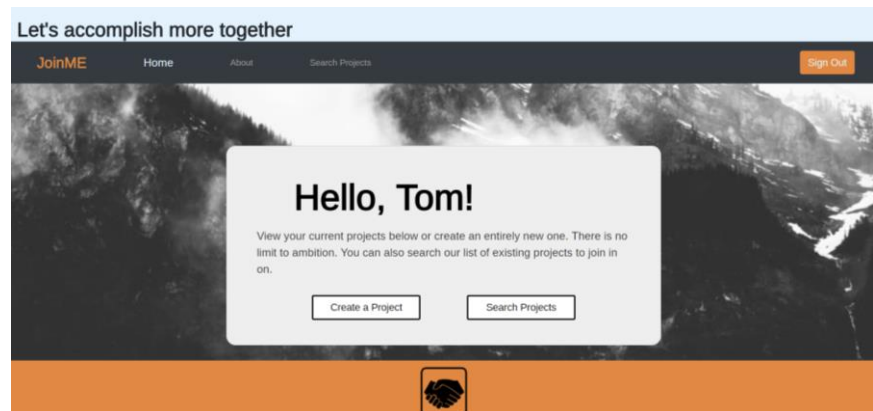
Steps:
Created by user
Belongs to Projects

```
JS steps.js x
11     type: DataTypes.TEXT,
12     allowNull: false,
13     len: [1]
14   },
15   Completed: {
16     type: DataTypes.BOOLEAN,
17     allowNull: false,
18     defaultValue: false
19   }
20 });
21
22
23 Step.associate = function(models) {
24   Step.belongsTo(models.Project, {
25     onDelete: "CASCADE",
26     foreignKey: {
27       allowNull: false
28     }
29   });
30 };
31
32 Step.afterValidate(function(step) {
33   if (step.Completed === "True") {
```

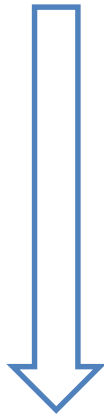
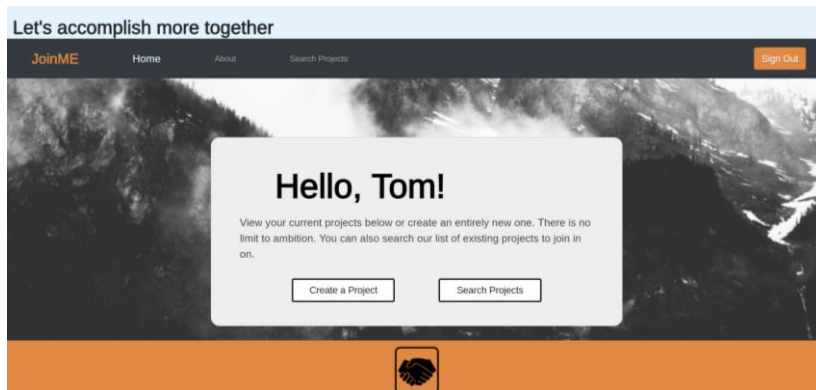
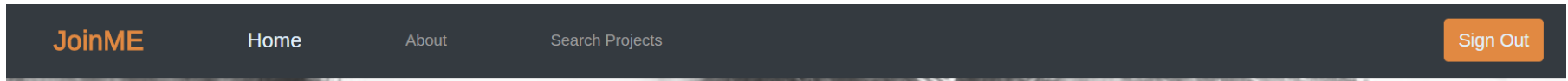
User Login



- Using Passport for middleware and auth
- Upon login user is presented with their page.
- User is inputted into the database



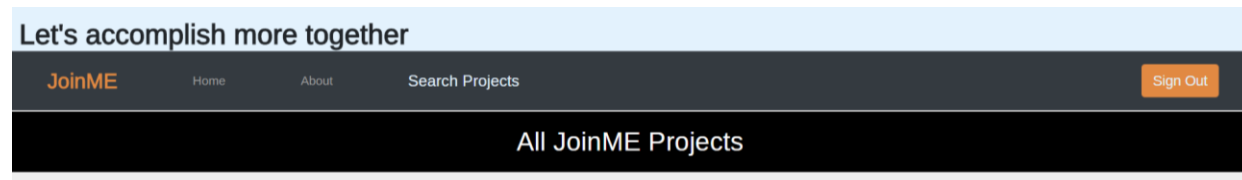
HTML Routes



- Home is users page
- Search projects is a summary of all user(s) projects:
 1. User can select to “Join”
- Create a project is a modal:
 1. User defines a new project with steps
- About is summary page

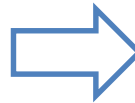
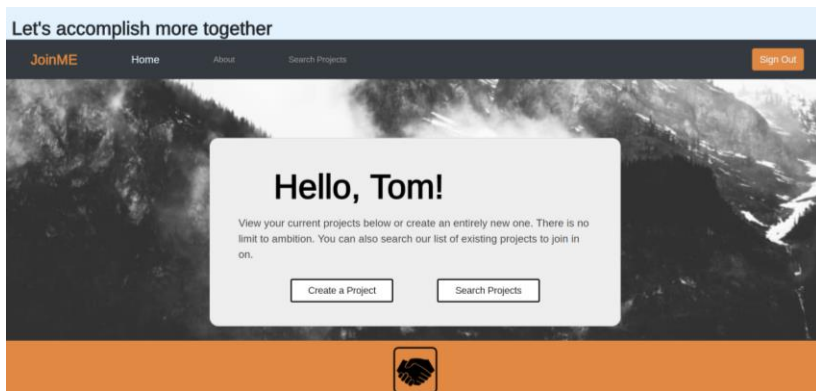


- Sign out



HTML Routes “continued”

- Create a project is a modal:
 1. User defines a new project with steps



Create your Project

Project Name:

Choose a Project Type

Options

Project Description:

Step One:

Step One Description:

Step Two:

Step Four:

Step Four Description:

Step Five:

Step Five Description:

Click "submit" and take that first step.

submit

close

API ROUTES

- User Routes

1. GET: All User
2. POST: New User

```
db.User.create({  
  name: req.body.name,  
  phone: req.body.phone,  
  email: req.body.email,  
  uid: req.body.uid
```

- Projects Routes

1. GET: All and GET User
2. POST: Project for User

```
db.Project.create({  
  Project_name: req.body.Project_name,  
  Project_type: req.body.Project_type,  
  Project_descripton: req.body.Project_descripton,  
  UserId: req.body.userId
```

- Steps Routes

1. GET: For Project
2. POST: For Project

```
db.Step.create({  
  ProjectId: req.body.ProjectId,  
  Step: req.body.Step,  
  Step_description: req.body.Step_description,  
  Completed: req.body.Completed,  
}).then(function (results) {  
  //  
  console.log("created step.");  
});
```



- Authentication middleware for Node
- Simplified relational db interop vs Google Sign In

```
passport.serializeUser(function(user, done) {
  done(null, user.id);
});

passport.deserializeUser(function(id, done) {
  User.findById(id).then(function(user) {
    if (user) {
      done(null, user.get());
    } else {
      done(user.errors, null);
    }
  });
});
```

Testing Summary

Travis CI:



- language: node_jsnode
- js: - "8"
- install:
- - npm install
- - npm install -g mocha
- - npm install -g chai

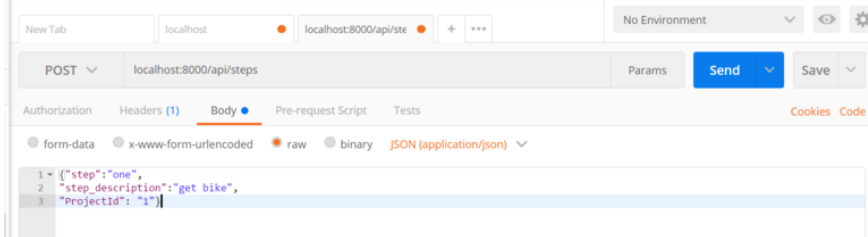
Test written:

```
var expect = require("chai").expect;
const testing = require("../routes/testing.js");

describe("CreateNewProjectButton", function() {
  it("should open the modal containing the project creation form when clicked", function() {
    expect(function() {
    })
  }
})

describe("Post user to user table", function() {
  it("it should post user to user database", function() {
    expect(function() {
      userPostWorked
    })
  }
})
})
```

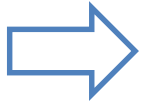
POSTMAN:



&&&& BRUTE FORCE!

Agenda

- Our motivation/summary
- Application's concept
- Technologies you used
- Design process
- DEMO!
- Future development



Design Process

- *Group Brainstorm...!*

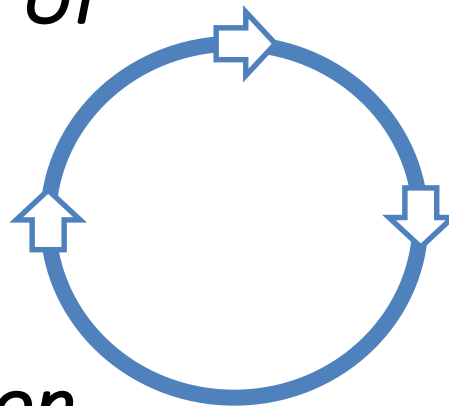
1. *Medical and Database and calendar*
2. *Projects and users with notification...*

- *Schema design*

Divided work:

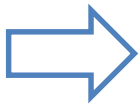
1. *Emma client js and UI*
2. *Paul, Tom UI, db*
3. *Chase Routes & js*
4. *Justin passport, js,
Model/route integration*

1. *Class*
2. *Slack*
3. *Git Hub*
4. *Google Hangouts*
5. *Testing*



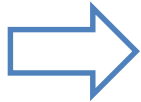
Agenda

- Our motivation/summary
- Application's concept
- Technologies you used
- Design process
- DEMO!
- Future development



Agenda

- Our motivation
- Application's concept
- Design process
- Technologies you used
- DEMO!
- Future development



Future development

- Filtered searches
- Sorting on page of searches
- Enhanced messaging
- Flexible project data manipulation by multiple users
- SMS messaging integration for tracking

Questions?