|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PROJECT DETAILS FORM: dev.rdolcegroup.com | | | | | |
| OVERVIEW | MERN INSTALLATION | | Primary Domains | | Location Paths |
| Client: React Server: MongoDB, Express, Node.js | | dev.rdolcegroup.com | | $NAS\web\_share\ rdolcegroup.com |
| http://rdolcegroup.localhost | | \Users\mrsdo\Sites\dreamhost\rdolcegroup\apps.rdolcegroup.com |
| GitHub | | <https://github.com/mrsdo/apps.rdolcegroup.com> |
| CONTACTS | Client or Sponsor | Budget | | | Team Communications Lead |
| R. Dolce Group Pine Timber Ridge Office  Fort Myers, FL, 33913 | $xxxx.xxx | | |  |
| REQUIREMENTS | Milestones | | | | |
| * Setup Local/Remote application structure, documentation, workflows, and processes (GIT/DEV) * Create an express-server environment with globals to manage imported packages * Research potential production environments [Vercel, Netlify, Gatsby Cloud] - Ongoing | | | | |
| TEAM | Lead Programmer/Developer | | | Team Members & Stakeholders | |
| Martinique Dolce xxx-xxx-9482  [marti.dolce@29signals.org](mailto:marti.dolce@29signals.org) | | |  | |
|  | |

| RESEARCH & PLANNING |
| --- |

| WORKFLOWS | |
| --- | --- |
| This section explores a bottom-up approach to the development of this application. Starting with the environment or production platform, its limitations, and how those limitations may impact the site's overall functionality and outcomes.  **Core specifications require:**   * Property Listings * Users * Roles * MongoDB Atlas with Express, React and Node.js * Private Routing requires an administrative role with authentication |  |

| SITE ARCHITECTURE |
| --- |

| Directories | |
| --- | --- |
| CLIENT(Front-end)/ | SERVER(Express Server Generated Backend)/ |

## Managing Permissions & Access Via Roles

Text

Description automatically generatedThe question of how to add required data on server launch may require the concept of seeding, similar to ruby. I'll need to do more research on seeding data using Mongoose and possibly JSON.

Maybe something like: <https://stackoverflow.com/questions/44427563/what-is-the-best-way-to-seed-data-in-nodejs-mongoose>

<https://github.com/mrsdo/react-user-onboarding/blob/main/server/app/models/role.model.js>

### Authentication with Private Routes

Preventing users from editing core collections will require implementing roles. In this case, the roles are Admin, Agent, and User.

The following is a summary of permissions/capabilities for assigned roles:

Admin: GET/POST/PUT/EDIT/DELETE – ALL collections data

Agent: GET/POST/PUT/EDIT – Assigned Listing(s) || Personal Account excluding role and Tenant associated with agent listing(s), View/Track Listing Lead(s)

Tenant: GET – Assigned Listing | EDIT - Personal Account

Users: GET –Listings | EDIT – Personal Account | Request additional information from Agent about listings | Post about Listing needs

## Adding Roles at Server Launch

Server.js adds roles to MongoDB on launch.

|  |
| --- |
| DATA: Roles are automatically added to MongoDB via Mongoose |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELDS | DATATYPE | NULL (Y/N) | KEY | EXTRA |
| RoleID | INT | N | PRI | Auto\_Inc |
| RoleName | STR | N | N | - |

|  |
| --- |
| ADMIN ROUTING: Roles are limited to administrative login | roleRouter.js |

|  |  |  |  |
| --- | --- | --- | --- |
| VERB | ROUTE | ACTION: RolesRouter ('/') | DESCRIPTION |
| GET | '/roles/list' | @role = Role.All | FindAll |
| POST | '/roles/new/' | @role = Role.new via form | .post + ${req.body.RoleName} |
| ADMIN REDIRECT | '/roles/success' | @role = Role.put(role\_params) + redirect | .redirect to Add\_new\_role\_success.js |
| GET / ID | '/roles/:id' | @role = Role.find(params[:id]) | FindbyID = a specific Role |
| PUT/PATCH/EDIT | '/roles/:id/update' | @role.update(role\_params) + flash | UpdatebyID with (async + promise) |
| DELETE / ID | '/roles/:id/delete' | @role.destroy(params[:id]) | DeleteByID + confirmation delete |

## Users and Associating with Roles using Mongoose.Schema.Types.ObjectId

SRC: <https://youtu.be/gswBLB0q9Xk> and <https://youtu.be/gswBLB0q9Xk?t=557>

Next, a user.model.js is created to include a roles association. The userRouter.js will also need to be updated to ensure a default role is added when a user registers an account.

Model Details: All users will have a default role of user. Only Admins, Moderators can edit a user role. Logged in users are redirected to a dashboard. The dashboard should contain partials for accessing: profiles or any applicable Listing data.

|  |
| --- |
| DATA: Users |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELDS | DATATYPE | NULL (Y/N) | KEY | EXTRA |
| UserID | INT | N | PRI | Auto\_Inc |
| RoleID | Int | N |  | Permission Level(Admin, Agent, Tenant,User) |
| Firstname | STR | N |  |  |
| Lastname | INT | N |  |  |
| Username | STR | N |  |  |
| Email | STR | N |  |  |
| Password | Password | N |  |  |
| Image (Portfolio) | STR | Y |  |  |
| BIO | STR | N |  |  |
| HomeAddress | STR | N |  | Permanent Home Address (If applicable) |
| ListingID | STR | Y |  | Associated Listing |
| Last Login | DATETIME | Y |  | Session / JWT |

|  |
| --- |
| ROUTING: userRouter.js |

|  |  |  |  |
| --- | --- | --- | --- |
| VERB | ROUTE | ACTION | DESCRIPTION (See roles) |
| GET | '/user/list' | @user = User.All | FindAll |
| POST | '/user/new/' | @user = User.new via form | .post + ${req.body.UserName} |
| ADMIN REDIRECT | '/user/success' | @user = User.put(user\_params) + redirect | .redirect to Add\_new\_user\_success.js |
| GET / ID | '/user/:id' | @user = User.find(params[:id]) | FindbyID = a specific User |
| PUT/PATCH/EDIT | '/user/:id/update' | @user.update(user\_params) + flash | UpdatebyID with (async + promise) |
| DELETE / ID | '/user/:id/delete' | @user.destroy(params[:id]) | DeleteByID + confirmation delete |

The userRouter.js will need to include a default role assignment, and field validations when a new user is created

### Testing in Postman

[Link to section e.g., Testing: Users & Roles]

|  |
| --- |
| PROJECT WORKFLOW: Planning, Development, Implementation |

| Job Number | Business Area Requirements & Description | Functional Area |
| --- | --- | --- |
| PORT-### | As [Role or Permission type], I need to [task name]  [bgd: #4F4B45) | Ex. How or where will the user access this feature? |
| PORT-100 | Setup Local/ Remote development structure, workflows and processes for DevOPs (GIT/DEV/UAT/PROD) | NAS & Dreamhost |
| PORT-200 | Provide Documentation/learning/knowledge base accessible via the NAS (media, audio, images, PDF, Ms Word, PPT) | Knowledge Base, Documentation – see Envato theme for KB |
| PORT-300 | Replicate via SSH + SSL to remote IP, then: NPM Install | Networking/Security |
| PORT-400 | Establish working directories for Local Dev, Remote Dev (Secured with SSH Logins) | NAS & Dreamhost |
| PORT-500 | Assign/verify IP's and Port to access via local network and remote via SFTP/SSH/HTTPS and domains | Networking/Security |
| PORT-600 | Add pages for: app.js, index.js, start.js, .env, user.htaccess, | Users, Authentication, Access Permissions, **GROUPS** |
| PORT-700 | Add Views for: Landing, About, Projects, Contact, registration, login, thank-you | Users, Authentication, Access Permissions, **GROUPS** |
| PORT-800 | Add theme UI and Models for: header.js, footer.js, tables, cards, grids, etc: (see envato elements theme) | Themes, Branding, |
| PORT-900 | Add color scheme, styles, content, images and media files; include branding | Marketing Requirement |
| PORT-1000 | Add branding theme and templates policies | Marketing Requirement |
| PORT-1100 | Landing page>About>Project Directory>Contact>Terms of Use>Privacy | UI |
| PORT-1200 | Express, install --save-dev nodemon, body-parser, EJS Theme template, express-validator, .env, Mongoose, Bootstrap, http-auth, animate.css, etc. (See create-react-app with CORS) | Middleware & Back tiers |

| PROJECT WORKFLOW SOURCE LIST |
| --- |

| Activity: | Details: | Status: |
| --- | --- | --- |
| PORT-100 | Setup Local/ Remote development structure, workflows and processes for DevOPs (GIT/DEV/UAT/PROD) | done |
|  | Add required packages for baseline and development environment support-globals.js | done |
|  | Add MongoDB connection with Async | done |
|  | Add models to create schemas with Mongoose for **roles**, users, listings, HOA's | partial |
|  | Update roles and users association (schema and routes) |  |
|  | Testing in Postman |  |

|  |  |  |
| --- | --- | --- |
| STYLE GUIDE: | | |
| Asset Type | Name & Description | Functional Area |
| Colors | [add swatches here] | Guide |
| Fonts |  | Guide |
| stylesheet |  | CSS |
| Logos |  | Images/branding |
| Copywrite |  | Policies |
| Terms of Use |  | Policies |
| Licensing | MIT, WordPress, Components, Elements, Images Sources and links | Policies |

|  |  |  |
| --- | --- | --- |
| USER EXPERIENCE BOARD: | | |
| Asset Type | Name & Description | Feature or Functional Area |
| app.js | [ | / |
|  | | |
| App.js | [ | / |
|  | | |
| Start.js | [ | / |
|  | | |
| .env | [ | / |
|  | | |
| user.htpasswd/htaccess |  | /config |
|  | | |
| footer | [ | /models |
|  | | |
| header | [ | /models |
|  | | |
| landingpage | [Landing page>About>Project Directory>Contact>Terms of Use>Privacy | /views |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| USER EXPERIENCE BOARD: | | | |
| Asset Type | Name & Description | Feature or Functional Area | |
| project-directory | [ | /views | |
|  | | | |
| contact | [ | /forms | |
|  | | | |
| termsofus | [ | /public | |
|  | | | |
| Privacy | [ | /public | |
|  | | | |
| About |  | | /views |
|  | | | |
|  |  |  | |
|  | | | |
|  |  |  | |
|  | | | |
|  | [ |  | |
|  | | | |
|  |  |  | |

## POSTMAN TESTING

This section provides testing details to ensure functionality for GET/POST/PUT/DELETE is validated and secure using role-based limits. Include screen captures to be used as video storyboard.

|  |
| --- |
| POSTMAN TESTING: Roles |

| Activity: | Description: | URL: |
| --- | --- | --- |
| GET |  |  |
| POST |  |  |
| PUT/PATCH BY ID |  |  |
| DELETE |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| POSTMAN TESTING: Users role-based limitations |

| Activity + role: | Description: | URL: |
| --- | --- | --- |
| GET |  |  |
| POST |  |  |
| PUT/PATCH BY ID |  |  |
| DELETE |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| PROJECT RESEARCH & RESOURCES LIST |

| Resource: | URL | FEATURE OR FUNCTION |
| --- | --- | --- |
| Animate.css | [Animate.css | A cross-browser library of CSS animations.](https://animate.style/) | JavaScript animation? |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |