29 . SIGNALS

Work Order Form

WORK ORDER NO.

EDITOR. APPROVED.

CLIYYYYMMDDHM

~MDOLCE12.01.2023 YES

© 2022 29Signals LLC. All Rights Reserved. Fort Myers, Florida • FL 33913 • (239) 309-9482 | build: 2.20230112.19:03:49

PROJECT DETAILS FORM:

OVERVIEW

Project Type

Primary Domains

Location Paths

Application: MERN Node.js Version

apps.rdolcegroup.com

$NAS\Node\Partition\workspace\projects\apps.rdolcegroup.com

\*.rdolcegroup.com

https://apps.rdolcegroup.com (Not avail. Yet)

172.xx.x.xxx

https://github.com/mrsdo/apps.rdolcegroup.com

CONTACTS

Client or Sponsor

Budget

Team Communications Lead

29 Signals, LLC

Pine Timber Ridge Office Fort Myers, FL, 33913

$NA

R. Dolce Group

`REQUIREMENTS

Milestones

* Setup Local Development structure, workflows and processes for DevOPs (GIT/DEV/UAT/PROD)
* Install NPM, Node.js, MongoDB using Brew
* Provide Documentation/learning/knowledge base accessible via the LOCAL NAS (media, audio, images, PDF, Ms Word, PPT)
* Replicate via SSH + SSL to remote IP, then: NPM Install
* Establish working directories for Local(Dev), UAT(Secured with Login), Prod(https://shegottablog.com)
* Assign/verify IP’s for each device on the network with custom virtual directory or alias
* Add pages for: app.js, index.js, start.js, .env, user.htaccess,
* Add Views for: Landing, About, Projects, Contact, registration, login, thank-you
* Add UI Models for: header.js, footer.js, tables, cards, grids, etc:
* Add color scheme, styles, content, images and media files; include branding
* Remote ISP Node.js with passenger (?)

TEAM

Lead Programmer/Developer

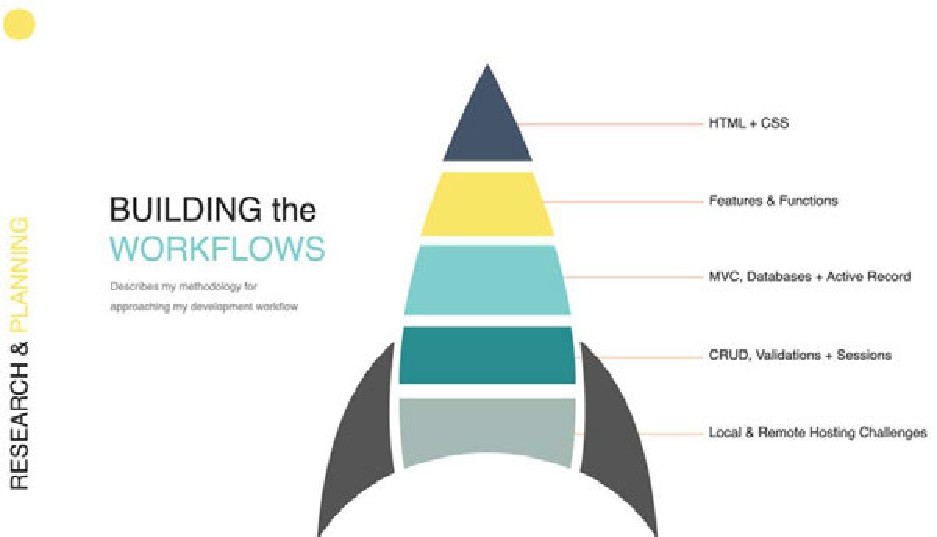
Team Members

Martinique Dolce

[marti.dolce@29signals.org](mailto:marti.dolce@29signals.org)

Alexa Johns | xxx-xxxx-0189 | [akjohns@rdolcegroup.com](mailto:akjohns@rdolcegroup.com)

Sam O’Conner | xxx-xxxx-2247 | [socon@29signals.org](mailto:socon@29signals.org)



*2*

*Dolce Management & Investment Properties, LLC • FL 33913 • (847) 226-2983*

RESEARCH & PLANNING:

WORKFLOWS

This section explores a bottom-up approach to the development of this application. Starting with the end- environment or production platform, it’s limitations and how those limitations may impact the overall functionality of the site and outcomes.

**Core specifications require:**

* Property Listings
* MongoDB Atlas with Express, React and Node.js

# 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, Moderator, and user.

Admin: GET/POST/PUT/EDIT/DELETE – ALL

Moderators: GET/POST/PUT/EDIT – Assigned Listing | Personal Account

Users: GET – Listings | EDIT – Personal Account

## 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 |

|  |  |  |  |
| --- | --- | --- | --- |
| VERB | ROUTE | ACTION: RolesRouter ('/') | DESCRIPTION |
| GET | '/roles' | @role = Role.All | .all((req, res, next) or .get((req, res) |
| GET | '/roles/new' | @role = Role.new via form | .post((req, res) + ${req.body.RoleName} |
| POST | '/roles' | @role = Role.put(role\_params) + redirect | Registration\_success.js |
| GET / ID | '/roles/:id' | @role = Role.find(params[:id]) | Find a specific Role |
| GET /EDIT ID | ‘/roles/:id/edit’ | @role = Role.update(params[:id]) + flash | Edit a specific Role |
| PUT/PATCH | '/roles/:id' | @role.update(role\_params) + flash | .put(async + promise) |
| DELETE | '/roles/:id' | @role.destroy(params[:id]) | .delete((req, res, next) + role.id |

*3*

*Dolce Management & Investment Properties, LLC • FL 33913 • (847) 226-2983*

ROUTING: Listings

VERB

ROUTE

ACTION

DESCRIPTION (See roles)

GET

‘/listings’

@listing = Listing.All

Display All

GET

‘/listings/new’

@listing = Listing.new

Add new with Form

POST

‘/listings’

@listing = Listing.put(listing\_params) + redirect

Add new Form Action

GET / ID

‘/listings/:id’

@listing = Listing.find(params[:id])

Show/Find by ID

GET /EDIT ID

‘/listings/:id/edit’

@listing = Listing.update(params[:id]) + flash

Edit by ID

PATCH

‘/listings/:id’

@listing.update(listing\_params) + flash

Update by ID

PUT

‘/listings/:id’

@listing.update(listing\_params) + flash

Replace by ID

DELETE

‘/listings/:id’

@listing.destroy(params[:id])

Delete by ID

DATA: Listings

FIELDS

DATATYPE

NULL (Y/N)

KEY

EXTRA

ID

INT

N

PRI

Auto\_Inc

Name

Int

N

N

Address

STR

N

N

-

City

INT

N

N

-

State

STR

N

N

-

Zip

STR

N

N

-

Community

STR

N

N

-