

## C868 – Software Capstone Project Summary

### Task 2 – Section C



Capstone Proposal Project Name: Project Dashboard Simplify

Student Name: Thuy Tien Nguyen

Table of Contents

## Contents

<i>Application Design and Testing</i> .....	4
<i>Design Document</i> .....	4
Class Design .....	4
UI Design .....	5
<i>Unit Test Plan</i> .....	6
Introduction .....	6
Purpose .....	6
Testing Environment .....	6
Test Cases .....	6
Test Plan .....	7
Items: .....	7
Features: .....	7
Deliverables: .....	8
Tasks: .....	8
Needs: .....	8
Pass/Fail Criteria: .....	8
Specifications .....	9
Procedures .....	10
Results .....	11
<i>C4. Source Code</i> .....	16

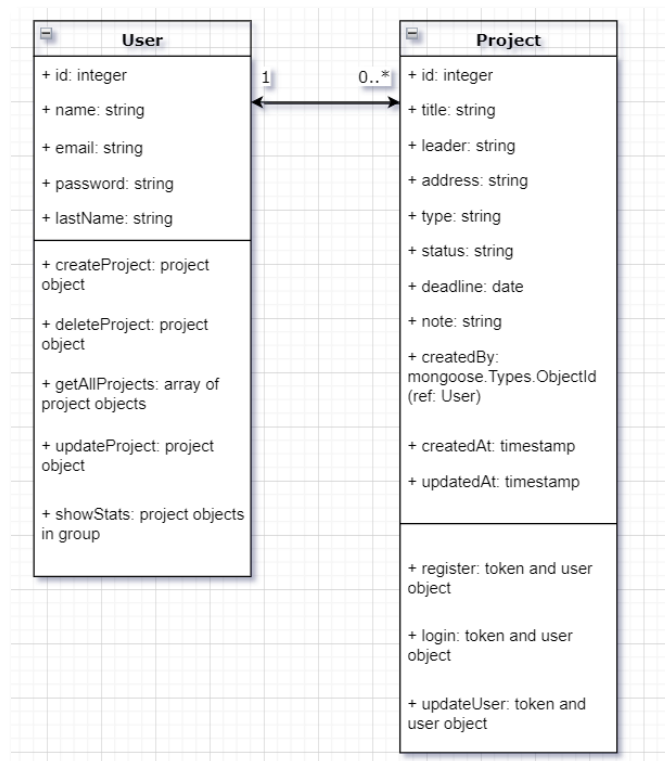
<b><i>C5. Link to Live Version .....</i></b>	<b><i>16</i></b>
<b><i>User Guide .....</i></b>	<b><i>17</i></b>
<b>Set Up and Run Applications for Maintenance Purposes.....</b>	<b>17</b>
Prerequisites .....	17
Installation .....	17
<b>Guide For Users .....</b>	<b>18</b>
❖ Register and Login:.....	<b>18</b>
❖ Dashboard Page .....	<b>19</b>
Status/Report Page .....	19
Create a New Project .....	20
View All Projects (Card View) .....	21
View All Projects (List View) .....	22
Update a Project .....	22
Delete a Project.....	24
Project Search and Filter .....	24
View a Single Project .....	25
View Profile .....	26
Pagination .....	26
Toggle Navigation Bar .....	27
Responsive Screen .....	28
<b><i>Sources .....</i></b>	<b><i>29</i></b>

## Application Design and Testing

### Design Document

#### Class Design

The application is a web-based application using MERN stack which utilizes NodeJS, Express server for the backend, and ReactJS framework for frontend backed by MongoDB database (a non-relational document database), this application embraces the MVC (Model-View-Controller) architecture. For this application model class design, there are two main model classes: the User model and the Project model. This is the one-to-many relationship between User and Project as a user can have many projects and a project can belong to at least one user as the following diagram:

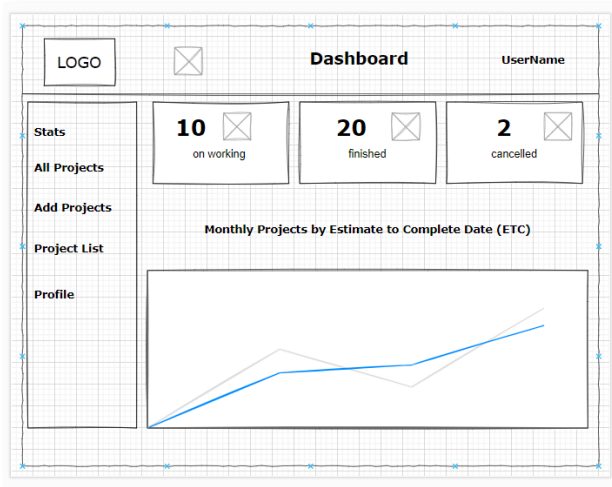


## UI Design

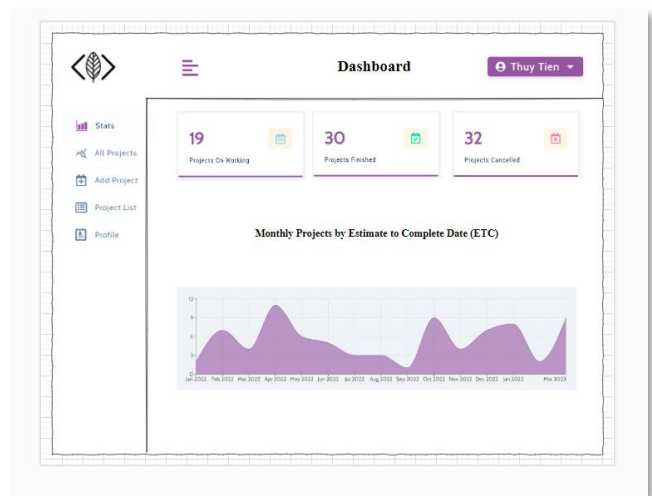
Below are the low-fidelity design and high-fidelity design of the dashboard main page. On the main page, the general report is already shown for the users to see the project status, such as the number of projects on working, finished, or canceled. The graph field is a visual line graphic for users to see how many projects they have each month in the cycle of a total of 12 months. To access other pages in the dashboard, users only need to access the main navigation bar on the left side which can be toggled to view the whole page of the dashboard when needed. By clicking on each item from the navigation bar, user can access their profile to update their credentials, add a new project, and view all projects either in card view or list view.

The following is the figure for low fidelity on the left side and high fidelity on the right side:

*Figure 1: Low Fidelity*



*Figure 2: High Fidelity*



# Unit Test Plan

## Introduction

This test plan outlines the steps and procedures for testing a RESTful API on the server side in the beginning stage built with Node.js and Express.js and using a MongoDB model.

RESTful APIs are a way of designing and implementing web services that follow the principles of Representational State Transfer (REST). REST is a set of architectural constraints that define how a client and server should communicate over the web.

The test environment used for this plan will be Postman and MongoDB compass to represent database document examples.

## Purpose

The purpose of this test plan is to ensure that the RESTful API is tested thoroughly and that any issues or defects are identified and addressed before the API is released to the next sprint planning based on our Agile methodology. The test plan will outline the testing environment, the test cases to be performed, and the reasons for their selection.

## Testing Environment

The testing environment for this project will be Postman. Postman is a popular API development and testing tool that allows you to make HTTP requests, validate responses, and automate API testing.

## Test Cases

The test cases to be performed in this project include:

- ✓ Functionality tests.
- ✓ Endpoint tests.
- ✓ CRUD (Create, Read, Update, Delete) tests.
- ✓ Performance tests.

- ✓ Security tests.

These test cases were selected to ensure that the API is tested thoroughly and that all aspects of the API are covered, including its functionality, endpoints, data management, performance, and security.

## Test Plan

### Items:

- ✓ Define the scope of the tests.
- ✓ Determine the testing environment.
- ✓ Identify the testing tools to be used.
- ✓ Define the test cases and test scenarios.
- ✓ Create test data.
- ✓ Define the testing schedule.
- ✓ Identify the testing team and their roles.

### Features:

- ✓ The functionality tests will test the API's functionality and ensure that it performs as expected.
- ✓ The endpoint tests will test the API's endpoints and ensure that they are accessible and responsive.
- ✓ The CRUD tests will test the API's ability to create, read, update, and delete data and ensure that it performs as expected.
- ✓ The performance tests will test the API's performance and ensure that it runs efficiently and meets the specified performance requirements.

- ✓ The security tests will test the API's security with JWT (JSON Web Tokens) and ensure that it is secure and meets the specified security requirements.

**Deliverables:**

- ✓ Test reports detailing the results of the tests.
- ✓ Test cases and test scenarios
- ✓ Test data
- ✓ Documentation outlining the testing process and results.

**Tasks:**

- ✓ Create the test cases and test scenarios.
- ✓ Set up the testing environment in Postman.
- ✓ Create test data.
- ✓ Run the tests in Postman.
- ✓ Analyze the test results.
- ✓ Document the testing process and results.

**Needs:**

- ✓ Access to the RESTful API
- ✓ Postman Testing Environment
- ✓ MongoDB compass UI

**Pass/Fail Criteria:**

- ✓ The tests will be considered a success if they pass without any errors or exceptions.

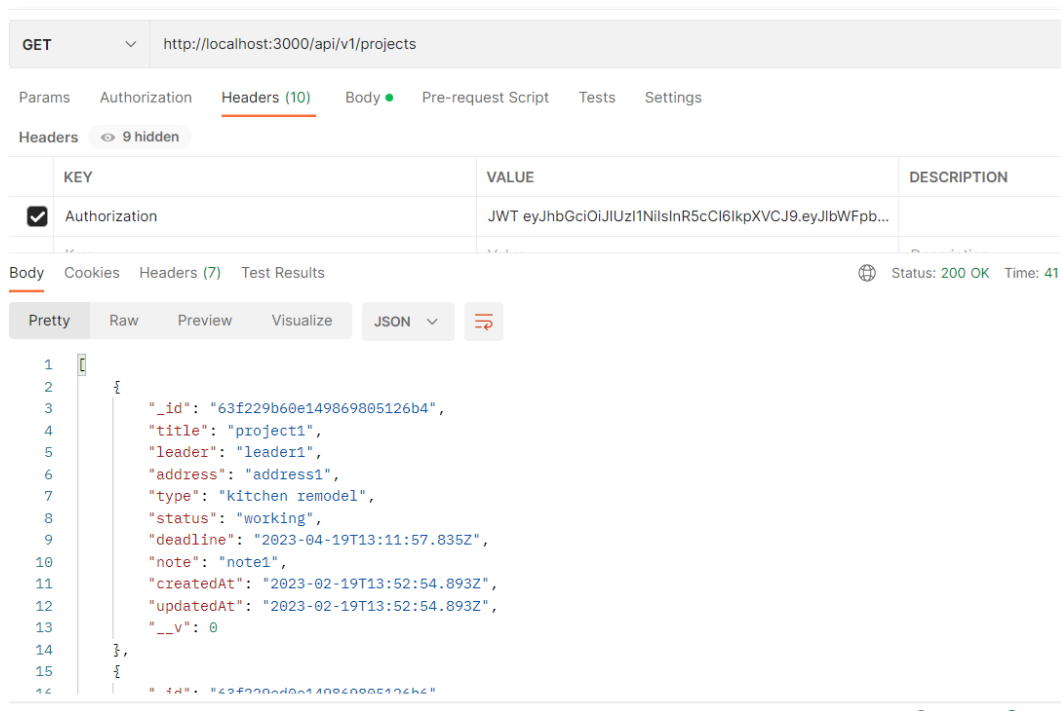


- ✓ If the tests fail, the cause of the failure will be analyzed, and a remediation strategy will be developed and implemented.
- ✓ Documentation of the testing process and results, including any failures, will be maintained.

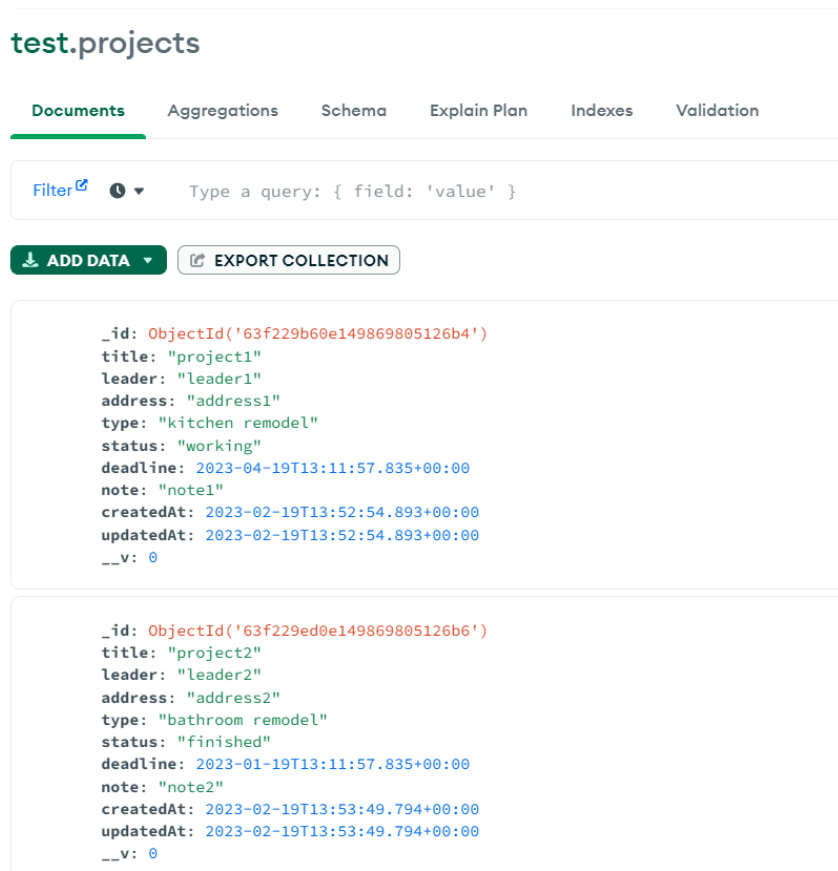
## Specifications

In Postman, we will create a request for a specific endpoint and add assertions to validate the response. For example, you can make a GET request to retrieve data and assert that the response code is 200 OK and that the response body contains the expected data.

Below is an example of getting data by making a GET request on Postman after a registered user logs in:



We also check if the data is also represented on the MongoDB database as below:



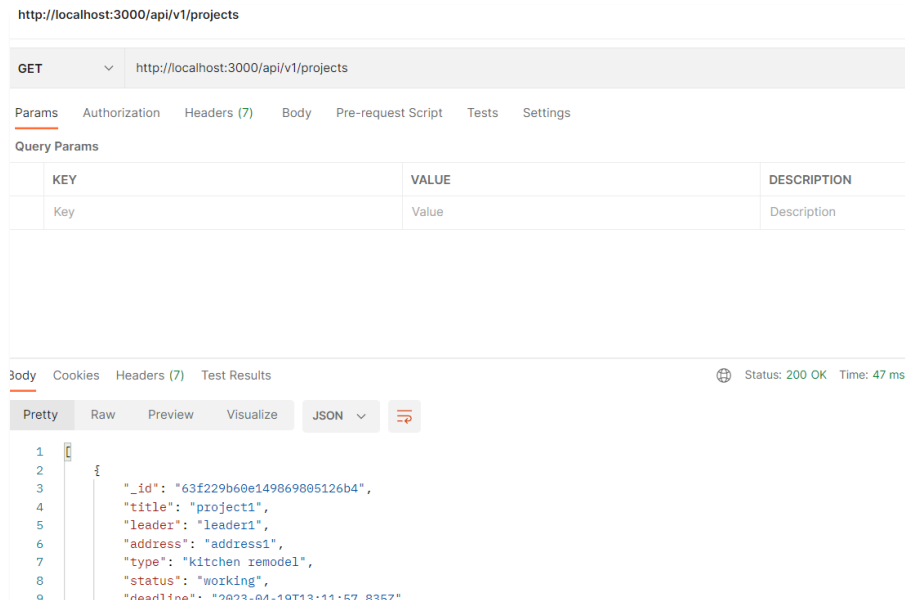
## Procedures

1. Create the test cases and test scenarios.
2. Set up the testing environment in Postman.
3. Create test data.
4. Run the tests in Postman.
5. Analyze the test results.
6. Document the testing process and results.

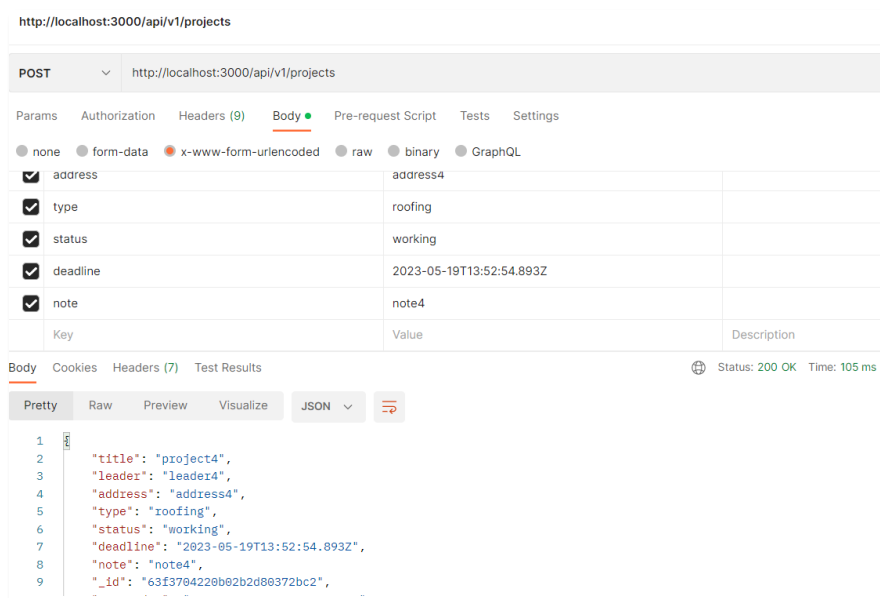
## Results

- ❖ Following are examples of CRUD (Create, Read, Update, Delete) tests on Postman:

### 1) GET all projects:



### 2) POST a project:



## 3) GET a specific project:

http://localhost:3000/api/v1/project/63f229ed0e149869805126b6

GET http://localhost:3000/api/v1/project/63f229ed0e149869805126b6

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
address	address4	
type	roofing	
status	working	
deadline	2023-05-19T13:52:54.893Z	
note	note4	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 43 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "63f229ed0e149869805126b6",
3   "title": "project2",
4   "leader": "leader2",
5   "address": "address2",
6   "type": "bathroom remodel",
7   "status": "finished",
8   "deadline": "2023-01-19T13:11:57.835Z",
9   "note": "note2",
10 }
```

## 4) PUT update a project:

http://localhost:3000/api/v1/project/63f229ed0e149869805126b6

PUT http://localhost:3000/api/v1/project/63f229ed0e149869805126b6

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

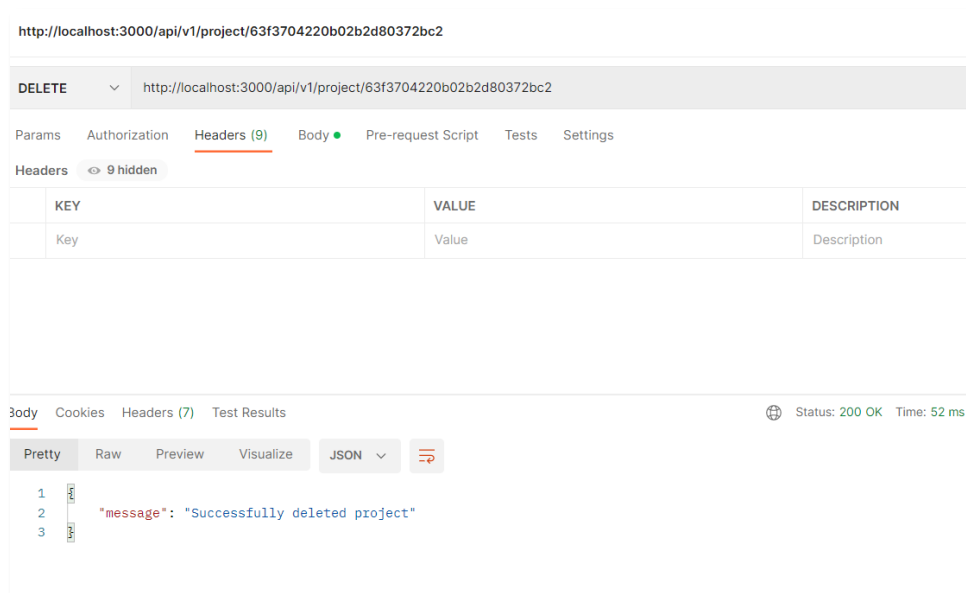
KEY	VALUE	DESCRIPTION
status	working	
deadline	2023-03-19T13:11:57.835Z	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 70 ms

Pretty Raw Preview Visualize JSON

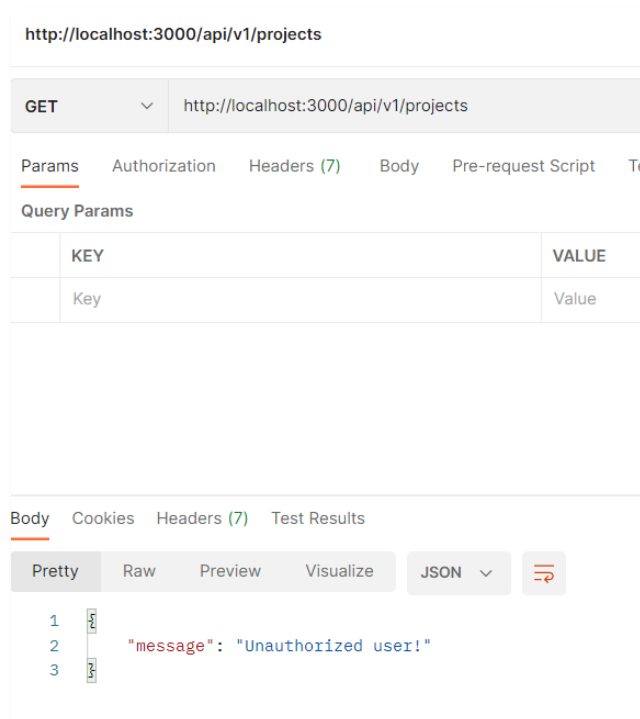
```
1 {
2   "_id": "63f229ed0e149869805126b6",
3   "title": "project2",
4   "leader": "leader2",
5   "address": "address2",
6   "type": "bathroom remodel",
7   "status": "working",
8   "deadline": "2023-03-19T13:11:57.835Z",
9   "note": "note2",
10 }
```

## 5) DELETE a project:



❖ Following are examples of security tests using JWT authorization on Postman:

## 1) Before the user logs in and sends a GET request, the message is shown below:



2) A new user can register as below:

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/api/v1/auth/register`. The 'Body' tab is selected, showing form data with the following fields:

KEY	VALUE
<input checked="" type="checkbox"/> name	user1
<input checked="" type="checkbox"/> email	user1@gmail.com
<input checked="" type="checkbox"/> password	test123
Key	Value

Below the form data, the 'Body' tab shows the JSON response:

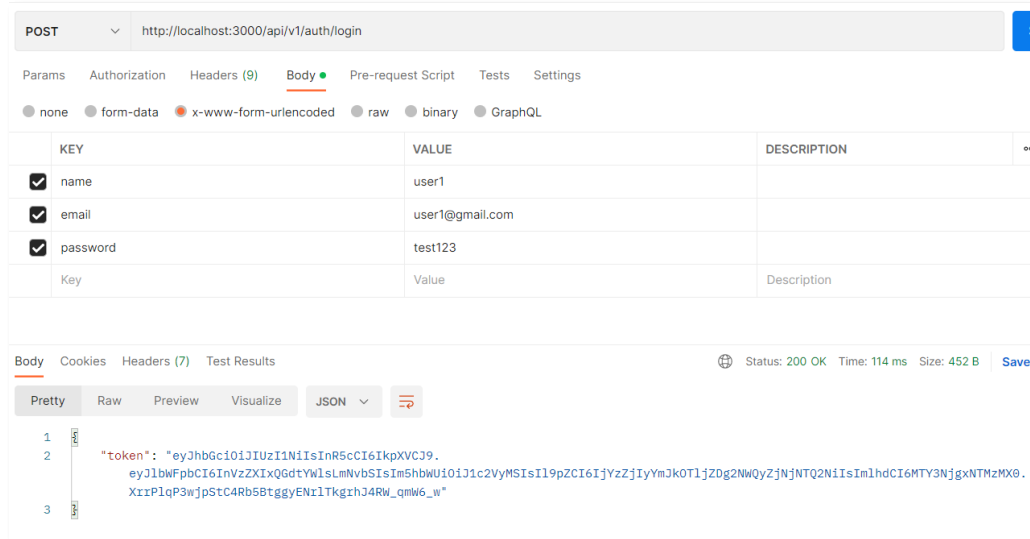
```
1 {
2   "name": "user1",
3   "email": "user1@gmail.com",
4   "_id": "63f22bbd99cd865d2f3c5466",
5   "__v": 0
6 }
```

This database object is also shown on the MongoDB database and the user's password becomes a serial hash number as below:

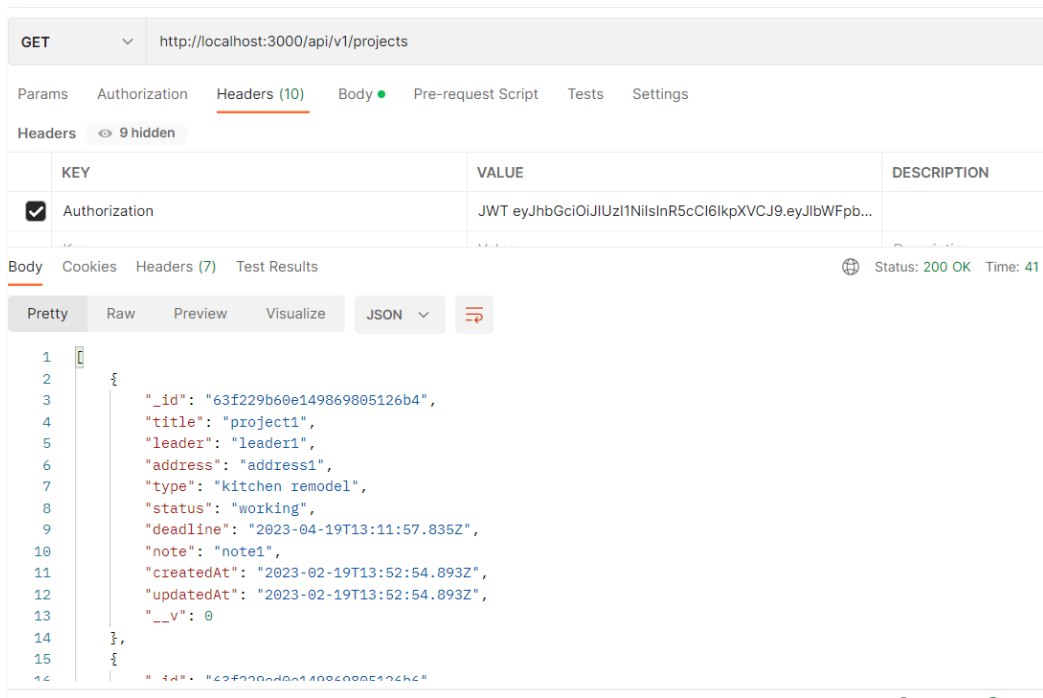
The screenshot shows the MongoDB Compass interface for the `test.users` collection. The 'Documents' tab is selected, and a single document is displayed:

```
_id: ObjectId('63f22bbd99cd865d2f3c5466')
name: "user1"
email: "user1@gmail.com"
password: "$2b$10$if5uGev50ydAP5nshgMT0.PESk8TYipbuhP.59JWvr1e5WYvMejHG"
__v: 0
```

3) A user logs in and gets JWT auth token:



4) Then, the user can use that token to be authorized to access the database as below:



## C4. Source Code

The production source code is published on GitHub by following this link:

[https://github.com/juniHub/MERN\\_ProjectDashboardSimplify\\_WGUCapstone](https://github.com/juniHub/MERN_ProjectDashboardSimplify_WGUCapstone)

The testing examples source code on Postman is published on GitHub by following this link:

[juniHub/TestPostmanProject \(github.com\) \(testing CRUD\)](#)

[juniHub/TestPostAuthUser \(github.com\) \(testing Auth\)](#)

## C5. Link to Live Version

The live version of this demo application is hosted on the Render platform as per the below link. However, I am using the **free plan** from the Render to demonstrate this web application, it is expected that the loading time is a little longer than normal, please be patient and reload again until you can see a full live webpage as shown below:

<https://projectdashboardsimplify.onrender.com>

For testing the mock-up database, use these credentials to log in:

Email: [testadmin@gmail.com](mailto:testadmin@gmail.com)

Password: **123abc**



## User Guide

### Set Up and Run Applications for Maintenance Purposes

To run the application locally you can follow these steps:

#### Prerequisites

The following system programs need to be installed:

- ✓ Visual Studio Code (latest version) (<https://code.visualstudio.com/>)
- ✓ NodeJS (latest version) (<https://nodejs.dev/en/>)
- ✓ Register to your MongoDB account, then follow the instructions to get your connection URL which will be used to set your private environment `.env` file.  
(<https://www.mongodb.com/>)

#### Installation

Use the below command on Visual Studio Code by following these steps to run the application locally.

1. Clone the repo on GitHub:

```
git clone https://github.com/juniHub/MERN_ProjectDashboardSimplify_WGUCapstone
```

2. Install dependency packages:

```
npm install
```

3. Create a `.env` file in the root folder and add the information below:

```
MONGO_URL=yourmongodbconnectionurl  
  
JWT_SECRET=yourjwtsecret  
  
JWT_LIFETIME=1d
```

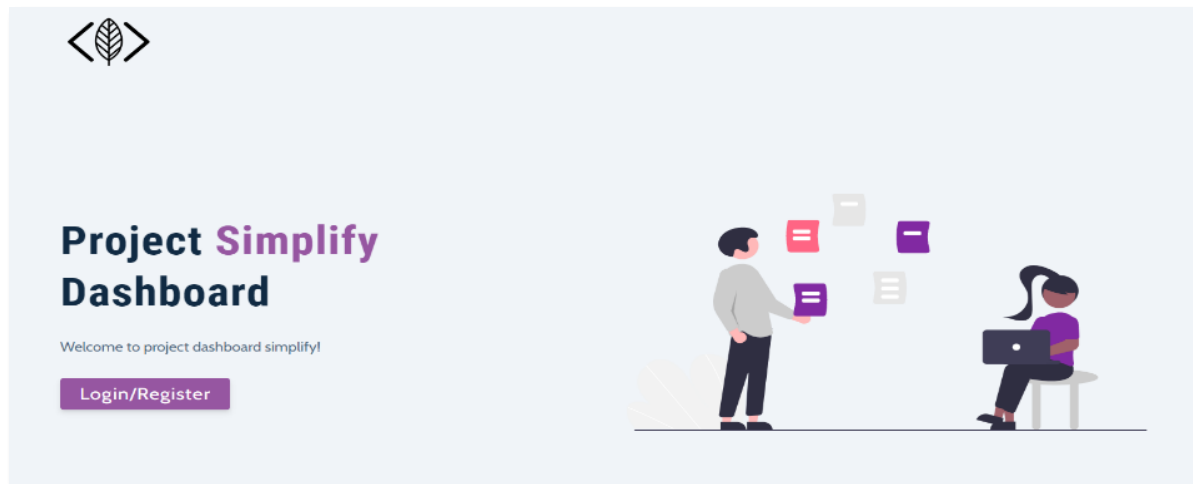
4. Run the project locally on <http://localhost:3000>

```
npm run start
```

## Guide For Users

The following steps will help users to start using the Project Dashboard Simplify application.

### ❖ Register and Login:



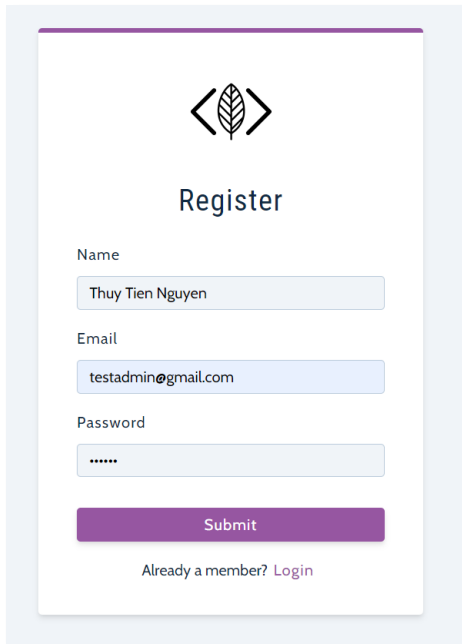
*Graphic Design from unDraw  
Source: <https://undraw.co/>*

As a first-time user, you need to register by clicking on “Register” to sign up with your credentials before you can log in.

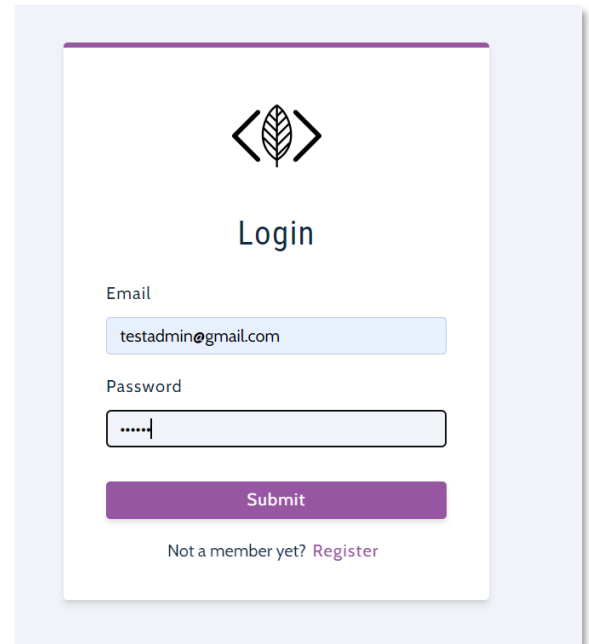
For testing the mock-up database to evaluate the project, click on the button “Login/Register” and enter the below credentials, then click on the “Submit” button. There is also an alert for every time users successfully or fail to log in or register.

Email: [testadmin@gmail.com](mailto:testadmin@gmail.com)

Password: **123abc**



The Register form features a logo at the top consisting of a leaf inside angle brackets. Below the logo is the title "Register". The form contains three input fields: "Name" with the value "Thuy Tien Nguyen", "Email" with the value "testadmin@gmail.com", and "Password" with masked characters ".....". A purple "Submit" button is positioned below the fields. At the bottom, there is a link "Already a member? Login".

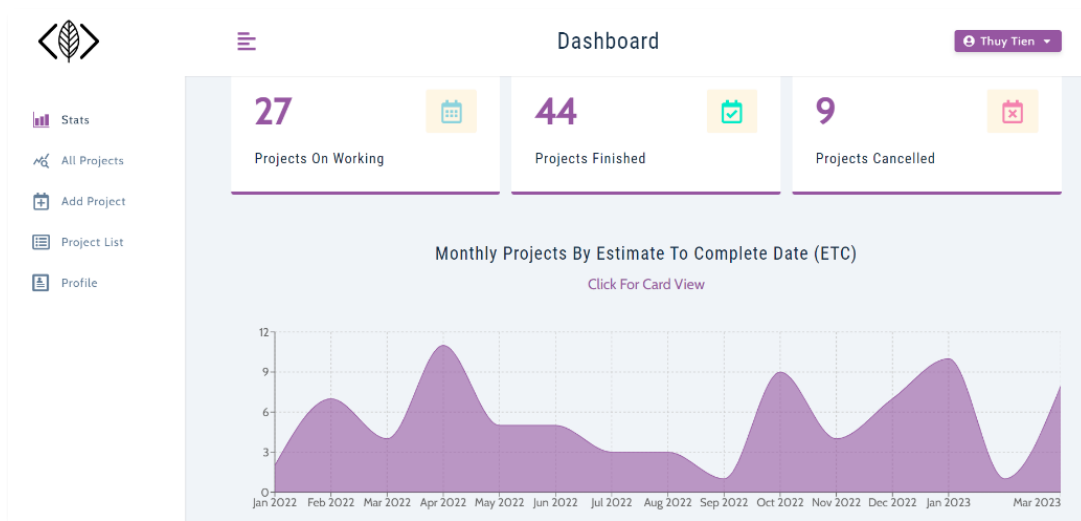


The Login form features the same logo as the Register form. Below the logo is the title "Login". The form contains two input fields: "Email" with the value "testadmin@gmail.com" and "Password" with masked characters ".....". A purple "Submit" button is positioned below the fields. At the bottom, there is a link "Not a member yet? Register".

## ❖ Dashboard Page

### Status/Report Page

By default, when a user logs into the dashboard successfully, at first, they can see a homepage which is also a status reporting page such as the total number of projects working, finished, or canceled. In addition, the user also can see the visual area chart which shows the total number of projects each month based on Estimate to Complete Date (ETC). Users can opt to see a card view for this number also.



### Create a New Project

On the main navigation bar on the left side, click on the “Add Project” item, this will be linked to a new page for a user to fill in all the information for the project as below figure. Click on the “Submit” button to create a new project or “Clear” to remove all information. If the user wants to cancel the process, just click on “Back to All Projects”. The user is required to fill in at least 3 information (title/customer name, address, leader/manager) to submit a new project. After a user successfully or fail to submit a new project, there is always an alert to notify the user about the process.

The screenshot shows the 'Add Project' form in the Project Dashboard. The dashboard header includes a logo, a menu icon, the title 'Dashboard', and a user profile 'Thuy Tien'. The left sidebar contains navigation items: Stats, All Projects, Add Project, Project List, and Profile. The main content area has a 'Back To All Projects' button and the 'Add Project' form. The form includes fields for Title/Customer Name, Address, Leader/Manager, Type (with a dropdown menu showing 'kitchen remodel'), Estimate To Complete (ETC) (with a date picker), Status (with a dropdown menu showing 'working'), and a large text area for Notes. At the bottom of the form are 'Submit' and 'Clear' buttons.

### View All Projects (Card View)

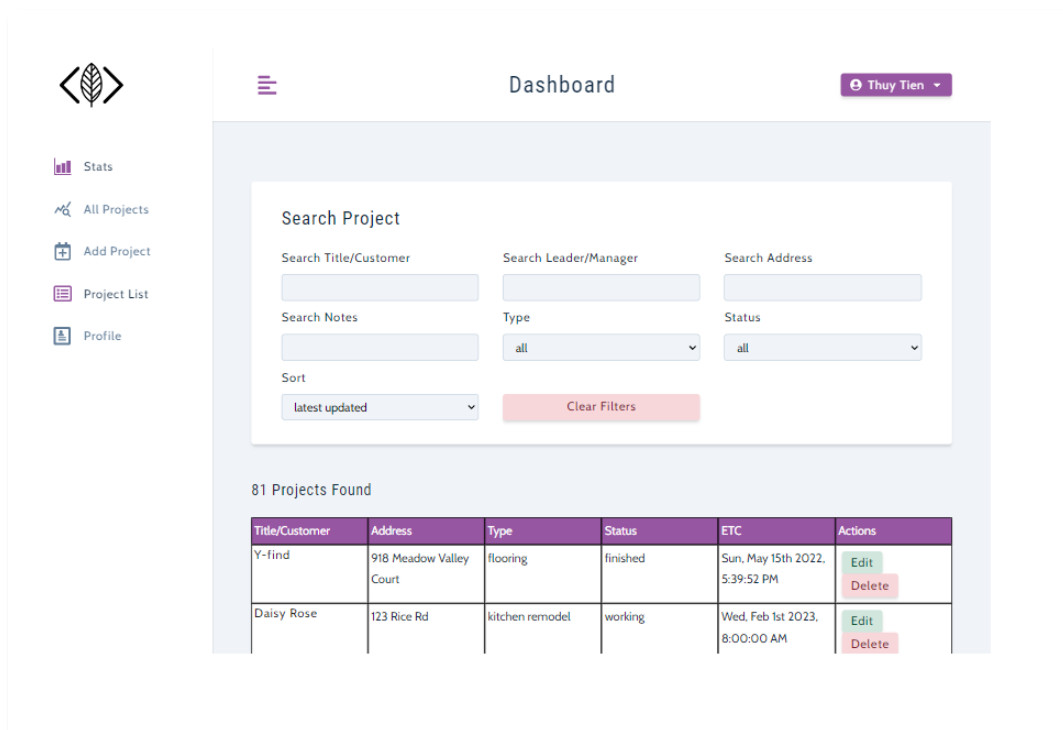
By clicking on the “All Projects” item on the navigation bar, the user will see all projects in card view format as shown below:

The screenshot shows the 'All Projects' card view in the Project Dashboard. The dashboard header and left sidebar are the same as in the previous screenshot. The main content area features a 'Search Project' section with filters for Search Title/Customer, Search Leader/Manager, Search Address, Search Notes, Type (dropdown menu showing 'all'), Status (dropdown menu showing 'all'), and Sort (dropdown menu showing 'latest updated'). A 'Clear Filters' button is also present. Below the search section, it states '81 Projects Found'. Two project cards are displayed: 'Y-Find' with address '918 Meadow Valley Court' and status 'Finished', and 'Viva' with address '88695 Cambridge Junction' and status 'Finished'. Each card shows the project type, ETC, and timestamp.

Project Name	Address	Status	Type	ETC	Timestamp
Y-Find	918 Meadow Valley Court	Finished	Flooring	Sun, May 15th	2022. 05:39:52 PM
Viva	88695 Cambridge Junction	Finished	Interior/Exterior Paintine	Thu, Apr 7th	2022. 02:54:37 PM

## View All Projects (List View)

By clicking on the “Project List” item on the navigation bar, the user can see all projects in table format as shown below:



The screenshot displays the 'Project List' view on a dashboard. The sidebar on the left contains navigation links: Stats, All Projects, Add Project, Project List (selected), and Profile. The main content area features a 'Search Project' section with the following filters:

- Search Title/Customer:
- Search Leader/Manager:
- Search Address:
- Search Notes:
- Type:
- Status:
- Sort:
- Clear Filters button

Below the search section, it indicates '81 Projects Found' and displays a table of project details:

Title/Customer	Address	Type	Status	ETC	Actions
Y-find	918 Meadow Valley Court	flooring	finished	Sun, May 15th 2022, 5:39:52 PM	<a href="#">Edit</a> <a href="#">Delete</a>
Daisy Rose	123 Rice Rd	kitchen remodel	working	Wed, Feb 1st 2023, 8:00:00 AM	<a href="#">Edit</a> <a href="#">Delete</a>

## Update a Project

To edit a project, the user can either click on the “Edit” button on the “All Projects” page or the “Project List” page, or the single project page as shown in the red circle below.

After that, a new page “Edit Project” will be shown as follows, user can start to edit information, then click on the “Submit” button to save updated information to the database:

Y

Y-Find  
Address: 918 Meadow Valley Court  
Finished

Type: Flooring

ETC: Sun, May 15th 2022, 05:39:52 PM

See Details Edit Delete

Back To All Projects

Y

Y-Find  
Address: 918 Meadow Valley Court  
Finished

Type: Flooring

ETC: Sun, May 15th 2022, 5:39:52 PM

Leader/Manager: Alta Kneafsey

Updated: Feb 5th, 2023, 1:39:19 PM

Note: Cum Sociis Natoque Penatibus Et Magnis Dis Parturient Montes, Nascetur Ridiculus Mus. Vivamus Vestibulum Sagittis Sapien. Cum Sociis Natoque Penatibus Et Magnis Dis Parturient Montes, Nascetur Ridiculus Mus.

Edit

Title/Customer	Address	Type	Status	ETC	Actions
Y-find	918 Meadow Valley Court	flooring	finished	Sun, May 15th 2022, 5:39:52 PM	<span>Edit</span> <span>Delete</span>
Daisy Rose	123 Rice Rd	kitchen remodel	working	Wed, Feb 1st 2023, 8:00:00 AM	<span>Edit</span> <span>Delete</span>

## Edit Project

Title/Customer Name

Y-find

Address

918 Meadow Valley Court

Leader/Manager

Alta Kneafsey

Type

flooring

Estimate To Complete (ETC)

mm/dd/yyyy --:-- --

Status

finished

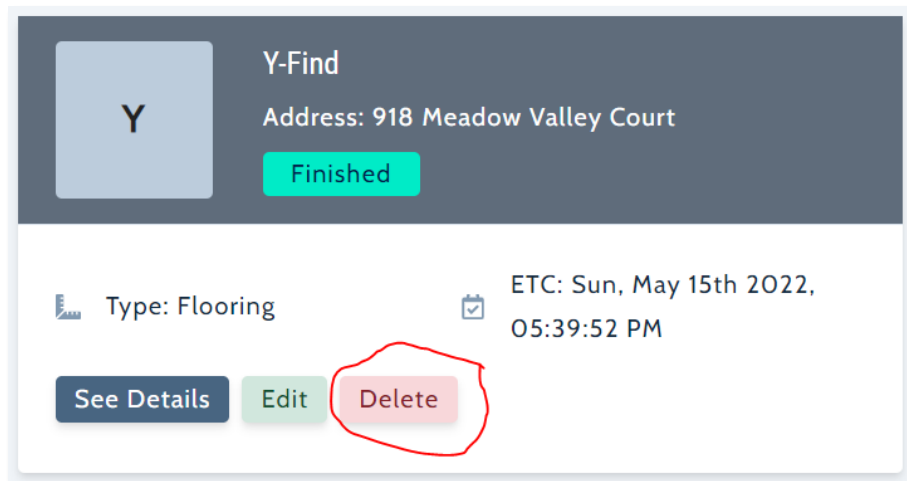
Notes

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vivamus vestibulum sagittis sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

SubmitClear

## Delete a Project

To delete a project, the user can either click on the “Delete” button on the “All Projects” page or the “Project List” page as shown in the red circle below, then a project will be removed from the database.



Title/Customer	Address	Type	Status	ETC	Actions
Y-find	918 Meadow Valley Court	flooring	finished	Sun, May 15th 2022, 5:39:52 PM	<a href="#">Edit</a> <a href="#">Delete</a>
Daisy Rose	123 Rice Rd	kitchen remodel	working	Wed, Feb 1st 2023, 8:00:00 AM	<a href="#">Edit</a> <a href="#">Delete</a>

## Project Search and Filter

One of the most important features of this application is the project search and filter. Therefore, a “Search Project” area will be shown above in both the “All Projects” page and “Project List” page for this application as screenshot below. Users have many options to search by project title/customer, project leader/manager and project address, and project notes to access information from the database quickly. Users can also sort



projects based on project status, project type, project updated date, and ETC date.

Clicking on the “Clear filters” button will reset all sort filter and search options:

**Search Project**

Search Title/Customer:

Search Leader/Manager:

Search Address:

Search Notes:

Type:

Status:

Sort:

[Clear Filters](#)

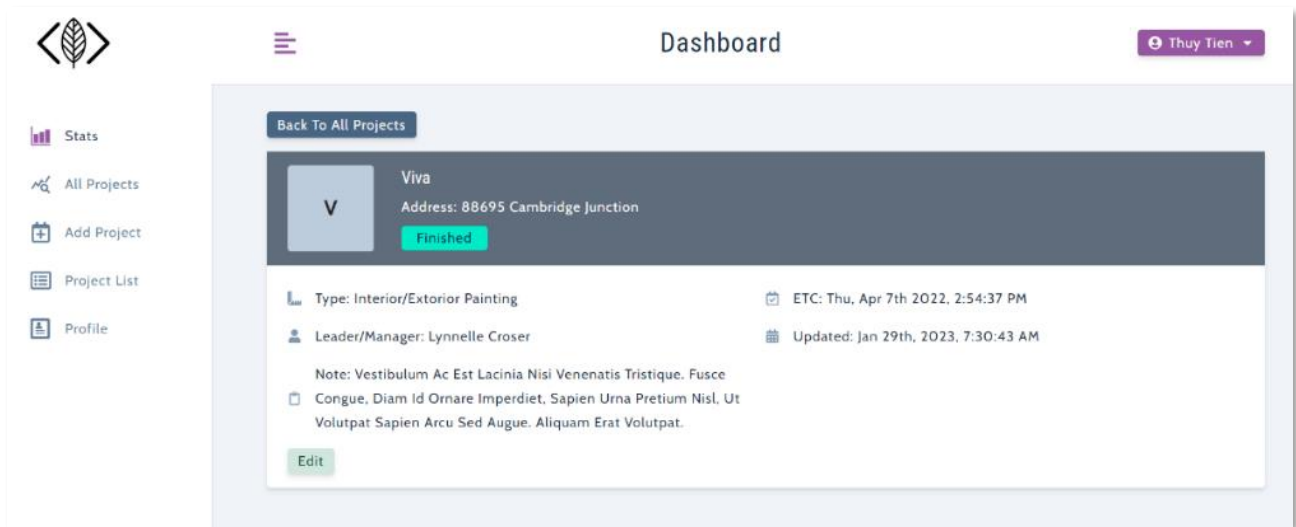
### View a Single Project

After clicking on the “See Details” button or a project title icon on the “All Projects” page or clicking on a project title on the “Project List” page as shown in the red circle below, the user can see more details of a single project they click on:

**Project Details Card:**

- Viva**
- Address: 88695 Cambridge Junction
- Finished**
- Type: Interior/Exterior Painting
- ETC: Thu, Apr 7th 2022, 02:54:37 PM
- [See Details](#) [Edit](#) [Delete](#)

Title/Customer	Address	Type
Y-find	918 Meadow Valley Court	flooring
Daisy Rose	123 Rice Rd	kitchen remodel
Viva	88695 Cambridge Junction	interior/exterior painting



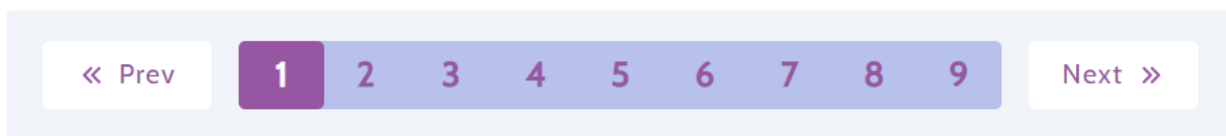
## View Profile

Users also can have a chance to modify their credentials by clicking on the “Profile” item in the navigation bar as shown below:

The screenshot shows a "Profile" page with a form to update user information. The form has four input fields: "Full Name" (containing "Thuy Tien Nguyen"), "Last Name" (containing "Nguyen"), "Email" (containing "testadmin@gmail.com"), and "Password" (empty). A purple "Save Changes" button is located to the right of the "Password" field.

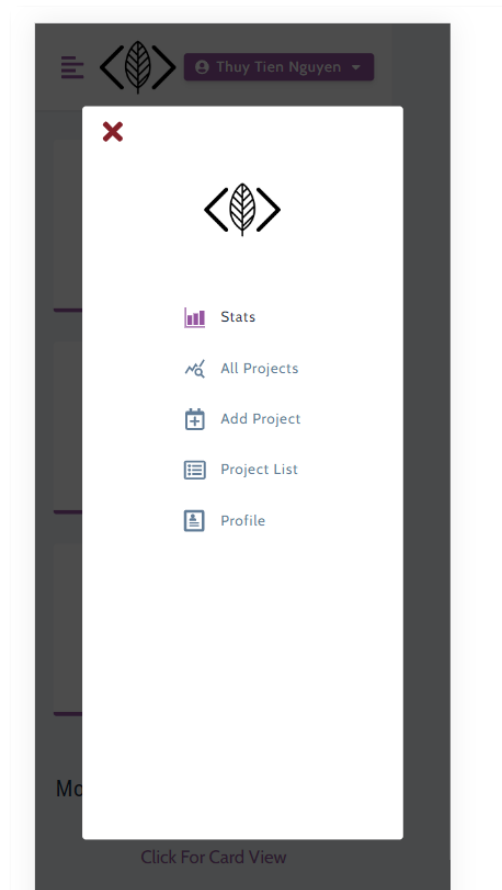
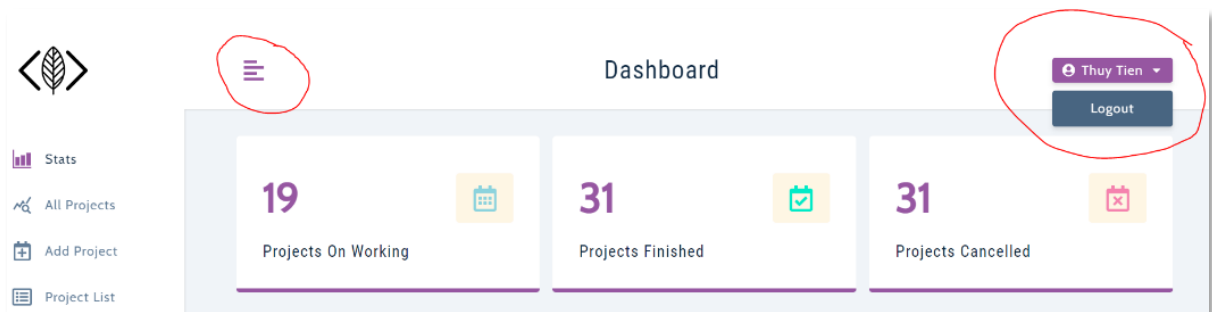
## Pagination

The bottom pagination shown on the “All Projects” and “Project List” pages will help the user to see all projects from the list.



## Toggle Navigation Bar

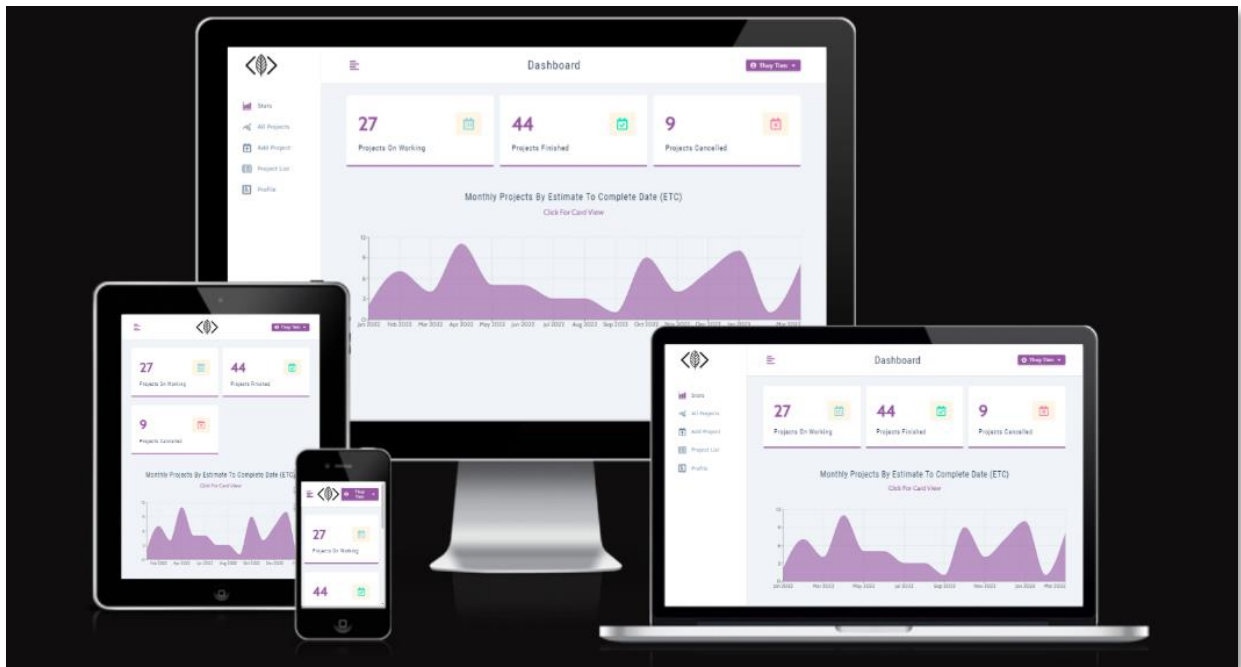
When a user clicks on the hamburger menu icon on the left side as shown on the red circle, the left side navigation bar can be hidden or shown. On the right side, when a user clicks on their username button, the “logout” button will be shown as below screenshot, then the user can log out from their account:



## Responsive Screen

One of the objectives of this application is a mobile-friendly UX/UI design so that users can access this application everywhere from their tablets or other mobile devices.

The demo screenshot as shown below demonstrates this requirement.



## Sources

unDraw. (n.d.). *Illustrations for Everyone*. Retrieved January 20, 2023, from <https://undraw.co/>