

Individual Project Task:

Develop an Information Logging Application called DocuTrack

When developers create applications, some employees find it very hard to use this application because they do not know how it works. Manifest Company wants you to create a simple information logging application called **DocuTrack** - this will enable developers to document how to use certain applications they developed for the company. Users can go on this application to read about the documentation for an application and maybe make a comment if they have any issues they encountered when using a certain company application. This application is very similar to my BlogApp project but must be branded in such a way that it reflects Documenting how to use an application as opposed to Blogging - see the requirements.

The DocuTrack application will follow this process.

- a. The DocuTrack app will have an Employee that creates documentation - this is the User Model. Please, no need to create an Employee Model again.
- b. A User will have a **Role, Department, Profile & Permissions**, and other fields based on the UserModule which you are already familiar with. Note: No need to create all these, they already exist in the starter kit provided.
- c. A **User** creates a Documentation for an application called **Document. This is a typical form with fields...**
- d. The Document must have a field called Subject, Description, **Status** Application Name, Type, and Category and all other nice to have fields.

- e. The **Type** of document can be categorized into Internal & External via a radio select button. I.e. The selection of type of document should be a radio select button.
- f. If a document is created as Internal, only Users with a Role called Manager or the Users that belong to that department (of the User that created the document) should be able to read/update or delete the document.
- g. The document can also be grouped into different **Application** such as Timesheet App, Expense Tracker App, Career App, Todo App e.t.c
- h. The Application field should be a select dropdown. So Users can select an Application they are writing the document for.
- i. A dedicated model should be created for Application drop-down selection such that Users can add different applications dynamically.
- j. The document can also be grouped into different **Category** such as 'How To', 'Common Issues', 'Database Schema', 'Model', 'Routes' e.t.c A documentation can be written for any of these Categories for that application.
- k. Again, the selection for Category should be a drop-down option.
- l. The fields that need to be filled should be dynamic based on the category selected. For instance, if the Category selected is Model, the form fields that need to be filled should be different from when a Category called Route is selected e.t.c you can also make certain fields required based on the category selected. For instance, for a

Model Documentation for an Application, you might want to include - upload model diagram. Whereas for Route documentation, you might want to include a text area field called valid Endpoints.

- m. When a document is created, the status to be changed to **Draft**. The User can publish this document later on.
- n. A User can change the status of the document they created. From Draft to Published or vice versa.
- o. If the role of a User is a Manager, they can read, edit, delete, or publish any document created by them or any User that belongs to same department as them.
- p. If the role of a User is a Team Member, they can read, edit, or **re-publish** the document created by themselves only. I.e. they cannot publish a document that is in draft created by another User, only role Manager can do that. They can only publish their own document.
- q. When a document is created, a **Comment** box should appear at the bottom of the document details page.
- r. The Comment box will have the general commenting pattern. I.e. Name of Commenter, subject, and the comment.
- s. Users can make comments on each document that has been published. Users should not be able to make a comment on an application that has not been published.

Important Information:

Documentation should be based on current business of the user. I.e. You should include the current business of user and department in the Document model such that when a User logs in, they can only see the Documents for their company or/and Department they belong to.

A theme has been provided, make sure your application is styled according to this theme. It must be cleaned and well presented. Ready to use in Production.

Start the application by developing the model diagram.

Create the CRUD for Document, Comment, Type & Category

Develop the association between these models appropriately.

Develop the logic for these associations based on the requirements above.

Develop the controller logic for these associations based on the requirements. Take your time with this but make sure your controller logic is clean and easy to read and understand.

You can also choose to develop a model for **Status** or any other Model to ease the CRUD process or put these fields Document model - this is up to you.

Ensure all incoming Data are validated before processing.

Contact Gospel on how to work with the Theme Provided as well as how to integrate the API JSON to your front end.

I want to see all API endpoints tested and integrated successfully. I want to see all form data validated successfully.

I want to see 5 major listing and details pages where applicable - **Users, Documents, Application, Type, and Category**, and I also want to see a **Dashboard** that provides a summary of what is going on in the company. I.e. Total documents created, documents by category, document by application, document by types, published documents, documents in draft e.t.c I also want to see the last five documents created in each of the categories and I want to be able to search or filter document list page.

Criteria for completion:-

- a) Ensure Project meets all features as stated in the project requirements.
- b) Ensure the API route is secured using the new method provided in the starter kit.
- c) Ensure all API errors are captured and logged.

- d) Ensure the application is API ready. I.e. can be integrated into any HTML/CSS template easily. All logic is contained in the controller.
- e) Ensure no broken links in Front End.
- f) Ensure function are in a try and catch block.
- g) Ensure all form fields coming to the backend are validated and cleaned up.
- h) Ensure no unnecessary files or commented code (clean up)
- i) Ensure functions have comments explaining what it does.
- j) Ensure API route / main routes are separated.
- k) Ensure API returns well-formatted JSON with code, data, and message.
- l) Ensure the login/logout works fine with normal login as well as Auto Login using the User Module app.

First Demo date (API / Backend - ready): expected Monday 13th of July, 2020

Second Demo date (API integrated with Front End - ready): expected 24th of July, 2020

Final Demo date: expected 30th of July, 2020

Sign off and Payment date: expected Friday 31st of July, 2020

Note: Your application will need to be signed off before getting paid for this month.

Note:

Prelude: The DocuTrack App will follow the usual UserModel pattern. I.e. the user that logs into the application would have attributes such as Role, Department, Profile, Current Business, Permissions e.t.c pre-populated. A sample starter kit has been created to ease your development. Please, feel free to use your own starter kit if you have an existing starter kit that works fine. Ensure your login/ log out works fine. However, you are not required to create the ability for a user to be created or updated in your final User Module. It will be automatically created or updated based on auto-login function. So comment out this route in your final deployment. I.e. remove the login access in the whitelist function inside server when completed.

Please, watch this video on how to use the User Module to log into your application:

<https://vimeo.com/433836636> password is **abc123**

Please download this starter kit for your application:

<https://microsolutions.s3.amazonaws.com/home/TraineesFolder/ManifestStarterKit.tar.gz>

Download the theme to use here - ensure you watch the video for how to use the theme too in documentation page of theme:

https://microsolutions.s3.amazonaws.com/home/TraineesFolder/keen_v1.4.9.zip