STELLAR 2020 TEST PROJECT – FRONTEND DEVELOPER
Task Management - Edit Task

TEST PROJECT DESCRIPTION

This test project is to create an "Edit Task" screen that forms part of a larger Task Management module. You should provide a means to open the edit task screen for each task by way of an unformatted list of tasks. No header, footer or menu is required; simply the edit task screen itself. You should be able to effectively demonstrate your ability to match the visuals provided alongside all documented frontend functionality, recognizing that no backend, i.e. server-side development, is required for this test project.

FRONTEND UI VISUALS

Add and Edit screens are attached in mobile and desktop formats in PNG and layered PSD formats, illustrating various functions defined in the specifications herein. While this test project is only for the Edit screen, the Add screen is included for reference because all its fields/functions are required on the Edit screen even if not depicted on the sample Edit screen visual. Layered PSDs and a style guide are also attached as well as the main screen visuals to give better context to the project. Remember that this test project is specifically the Edit screen, not the main screen, however a means to open the Edit screen for each task in an unformatted list of tasks is required.

TECHNICAL REQUIREMENTS

Required Languages/Frameworks (additional languages/frameworks may be used and are at the discretion of the frontend developer provided the required languages/frameworks are still used):
- HTML5
- SASS
- CSS3
- JavaScript
- React

- Bootstrap 4
- Reactstrap
- Redux
- Axios
- Formik
- Yup
- Font Awesome
- React-bootstrap-table2 ***

Cross-Browser Compatibility:
- Chrome 32+
- Firefox 24+
- IE 10+ ***
- Edge
- Safari 6+
- Opera 2014+

Cross-Device Compatibility:
- Windows 10
- Windows 10 Mobile
- Mac OS X 10.9+
- Android 4.4+
- iOS 7+

Technical Considerations:
- No backend required, strictly frontend, should be able to use locally without any server by simply opening in the browser.
- No usage of images; icons must use Font Awesome.
- UI should fill full screen width and be responsive between all Bootstrap 4 breakpoints.
- All data should be dummy and self-explanatory of what the data represents, e.g. "John Smith" as a name, "Module 1 – Record 7" as an associated record, "/[module]/edit/[id]" as a link to view/edit an associated record
- Dummy data should properly demonstrate not only the variations in data but also its completeness since these should be "existing" tasks which are all different, i.e. don't just copy the same data to 10 tasks because it doesn't effectively demonstrate the different capabilities for display
- SCRUD using RESTful API and JSON objects/arrays
    - Record searches via GET/POST (POST due to size limitations on GET requests)
    - Record creations via POST
    - Record reads via GET
    - Record updates via PUT
    - Record deletions via DELETE
    - All auto-completes, searches, etc. should make use of the record search capability

  - All API endpoints should be static .json files containing dummy data/responses for the purposes of this test project that could be swapped with true backend feeds
- Websocket implementations ***
  - If a single record has been updated while on the edit page, a modal should display to the user saying that the record has changed and asking if they would like to refresh to see the newest changes.
  - In any data tables, the table should automatically update after CUDs with filtering and sorting still applied
    - As an acceptable alternative, a "Live" checkbox which would enable auto-refreshing of the table every 60 seconds (60 being a system variable)

FUNCTIONAL SPECIFICATIONS

Epic: I can use an Edit Tasks page to edit tasks and view an audit log that shows all previously made edits to tasks.

Stories:

1. Audit Log

Description: The audit log will display all changes that are made to a task using a [Timestamp] [Who] changed [Field] from [Old Value] to [New Value] format. If no changes have been made the audit log will display a message stating, "No changes have been made."

Acceptance Criteria:

- I can see an audit log when I am on the edit task screen.
- I can see the audit log records in a [Timestamp] [Who] changed [Field] from [Old Value] to [New Value] format.
- The audit log displays "No changes have been made." if no changes have been made.
- I can scroll, paginate, sort, search, re-order columns, export and print audit log records by way of a data table. ***

2. Editable Fields

Description: Ability to make changes to all fields on a task including, but not limited to: Creator, Requestor Type (defaults to Customer, dropdown also shows Internal), Requestor (see #3), Department, Status, Category (see #5), Requested By Date, Requested By Time, Due Date, Due Time, and Associated Record (see #4). Use the attached "Add Task" and "Edit Task" visuals for the template of the edit screen and full list of fields and functions required. All task fields should be editable. Refer to the stories

3, 4, and 5 for information on the set up of the Requestor, Associated Record, and Category fields respectively.

Acceptance Criteria:

- I can use the Edit Task screen to edit all fields on a task as depicted on the Add Task and Edit Task visuals.
- I cannot save changes without passing the validation of required fields.
- I can optionally upload multiple files to attach to the Task as depicted on the Add Task visual.

3. Requestor Field

Description: When editing the Requestor field, if the Requestor Type is set to Internal, they are presented with a list of all employees/back end users. From the Requestor menu, I can search by any or all of the following fields: Customer ID, First Name, Last Name, Slip and/or Boat. When the user searches via one or more of these fields, they are presented with an autocomplete list of possible choices. These choices will display one option per boat per customer record, and will display the Customer ID, First Name, Last Name, Slip, and Boat. If there is no boat associated with a customer record, the boat field will appear as N/A. Use the attached Add Task visuals to determine the layout of the requestor field as well as the set-up of the auto complete dropdown.

Acceptance Criteria:

- When I select Internal as the Requestor Type, I can edit the Requestor field by searching for employees/back end users with an auto-complete list of possible choices.
- When I select Customer as the Requestor Type, I can edit the Requestor field by searching for customers by Customer ID, First Name, Last Name, Slip and/or Boat and see an autocomplete list of possible choices, represented as one choice per boat per customer.
- I can see that Customer records that contain no Boat list the boat field as N/A.
- When I select a Customer as the Requestor, the Slip and Boat fields will auto-populate with the values of the option I selected, if not empty.

4. Associated Record Field

Description: When editing the Associated Record field, they can select records from other modules that can be attached to the task they are currently editing. This will appear as an auto-complete drop down that provides them with options for records they can associate the task with.

Acceptance Criteria:

- I can use the Associated Record auto complete field to first select a module then a record within the module to attach to the task.
- I can add multiple Associated Records and remove Associated Records I previously added.

- I have a shortcut link to View/Edit where an Associated Record has been added.

5. Category Field

Description: When editing the Category field, a dropdown appears that lists all the Categories and any Categories with sub-categories (or categories that are of a lower hierarchy) can be found through an additional dropdown. Within each sub-category, any sub-categories that have more sub-categories of a lower hierarchy can be accessed via more dropdowns, so on and so forth, until you reach a sub-category with no sub-categories of a lower hierarchy. Users can then select the Category from the dropdown at the lowest hierarchy or Other at any level the hierarchy. At the bottom of all dropdowns is an Other option that will be hard-coded and not presented as an option via JSON/the backend.

Acceptance Criteria:

- I can edit the Category field from the edit tasks menu.
- When I edit the Category field, I can see a dropdown list that displays all categories and any categories with sub-categories (or categories that are of a lower hierarchy) can be found by clicking on a category that is indicated to be a dropdown. Sub-categories with sub-categories of a lower hierarchy also have dropdowns to display further sub-categories, so on and so forth, until there are no hierarchies below the one I selected.
- At the bottom of all Category and sub-category dropdowns, I can see an "Other" option.
- If I select an "Other" option in any level of the category hierarchy, I am required to populate a field containing the Nature of the Task.

*** Denotes an Extra Credit item that is not required for the test project but demonstrates additional skill.