UITableView Editing-06

In this assignment you will implement editing mode for the table view in your HomeOwner app.

1. Ensure you commit and push, then submit your commit ID for Lab 6 before moving on with this lab. Also, ensure you create a new repository on GitHub and connect it to your project. We will continue to work on your HomeOwner app for this lab.

2. Create a new repository on GitHub for this lab and open your HomeOwner app (note: ensure you committed and submitted the commit ID for lab 6 before moving onto this lab).

3. Go to your source control navigator and remove the remote for lab 6, add your remote for lab 7, and perform a push to ensure everything is setup.

4. Open Main.storyboard. From the object library, drag a View to the very top of the table view, above the prototype cell. This will add the view as a header view for the table view. Resize the height of this view to be about 60 points.

5. Now drag two Buttons from the object library to the header view and change their text to say "Add" and "Edit".

6. Create an action for each button called "addNewItem" and "toggleEditingMode".

7. Select both of the buttons and open the Auto Layout Align menu. Select Vertically in Container with a constant of 0. Make sure Update Frames is set to None, and then click Add 2 Constraints.

8. Open the Add New Constraints menu and add a leading constraint of 8, a trailing constraint of 8, and check Equal Widths. Make sure the values for the leading and trailing constraints save after you have typed them; sometimes the values do not save, so it can be a bit tricky. When you have done that, click Add 4 Constraints.

9. Implement your toggleEditingMode method by:
   a) Creating an if statement to check if the view controller is in editing mode
   b) If it is in editing mode, set the isEditing property to false, if it is not in editing mode, set the isEditing property to true
   c) You should set the isEditing property on the view controller itself. To set the isEditing property for a view controller, you call the method setEditing(_:animated:)
   d) Changing the text of the button to say "Edit" if isEditing is false and "Done" if isEditing is true

10. Implement your addNewItem method by:
    a) Creating a new item using your ItemStore method createItem
    b) Using an if statement to get the index for the new item and put it in a temporary variable
    c) Creating an IndexPath
    d) Use your table view's insertRows to insert a new row

11. Now that you have the ability to add rows and items, you no longer need the code that puts five random items into the store (this was done in your ItemStore class initializer). Remove the code you wrote to create 5 items.

12. In ItemStore.swift, implement a new method to remove a specific item.

13. In ItemsViewController.swift, implement tableView(_:commit:forRow:) by:
    a) Adding the method to your class
    b) Calling your new remove method in ItemStore to remove the right object
    c) Confirming the row deletion by calling the method deleteRows(at:with:) on the table view

14. In ItemStore.swift, implement a new method to move an item to a new index by:
    a) Creating a method called moveItem that take two parameters, one for the from index and one for the to index
    b) Creating a temporary variable to hold the item being moved
    c) Removing the item from the array
    d) Adding the item you stored in the temporary variable back into the array at the specified index

15. In ItemsViewController.swift, implement tableView(_:moveRowAt:to:) by:
    a) Adding the method to your class
    b) Call your new move item method in ItemStore to move the item

16. You are going to use a UIAlertController to confirm the deletion of items. You will use the .actionSheet style because the purpose of the alert is to confirm or cancel a possibly destructive action.

17. Update your tableView(_:commit:forRow:) implementation to include a UIAlertController by:
    a) Creating an instance of UIAlertController with the appropriate title, message, and style
    b) Creating a cancel action that does nothing when the user presses cancel
    c) Creating a delete action that runs the code you wrote to remove the item from the ItemStore and also from the table view
    d) Presenting the UIAlertController modally

Final App (In Editing Mode)