



Front-end React

Practice Exam

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	7/1/2019

Hanoi, mm/yyyy

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	30/May/2019	Create a new assignment	Create new	DieuNT1	VinhNV
2	07/Jun/2019	Update Fsoft Template	Update	DieuNT1	VinhNV

Contents

Customer Manager	4
Problem Description:	Error! Bookmark not defined.
Project Overview:	4
General functionalities:	4
Frontend Specs.....	4
Backend Specs	9
Login:.....	10
Logout:.....	10
Update User (Settings page)	10
List All Customers	10
List All Customers by page	10
List Customer by Id	10
Edit Customer By Id	11
Delete Customer By Id	11
Users (for authentication)	11
Customers.....	11
Customers By page.....	11
Errors and Status Codes	12
Other status codes:	12



CODE:	FEFW_React.Practice.T101
TYPE:	Exam 01
LOC:	N/A
DURATION:	N/A

Customer Manager

Project Overview:

You have to implement a Customer Management Web app using React.

General functionalities:

- Authenticate users (login pages and logout button on header)
- CRUD Customer
- GET and display paginated lists of Customer
- Changer user's email and setting in Settings page (**Optional**)

Frontend Specs

Using the API

See Backend Specs below.

Routing Guidelines

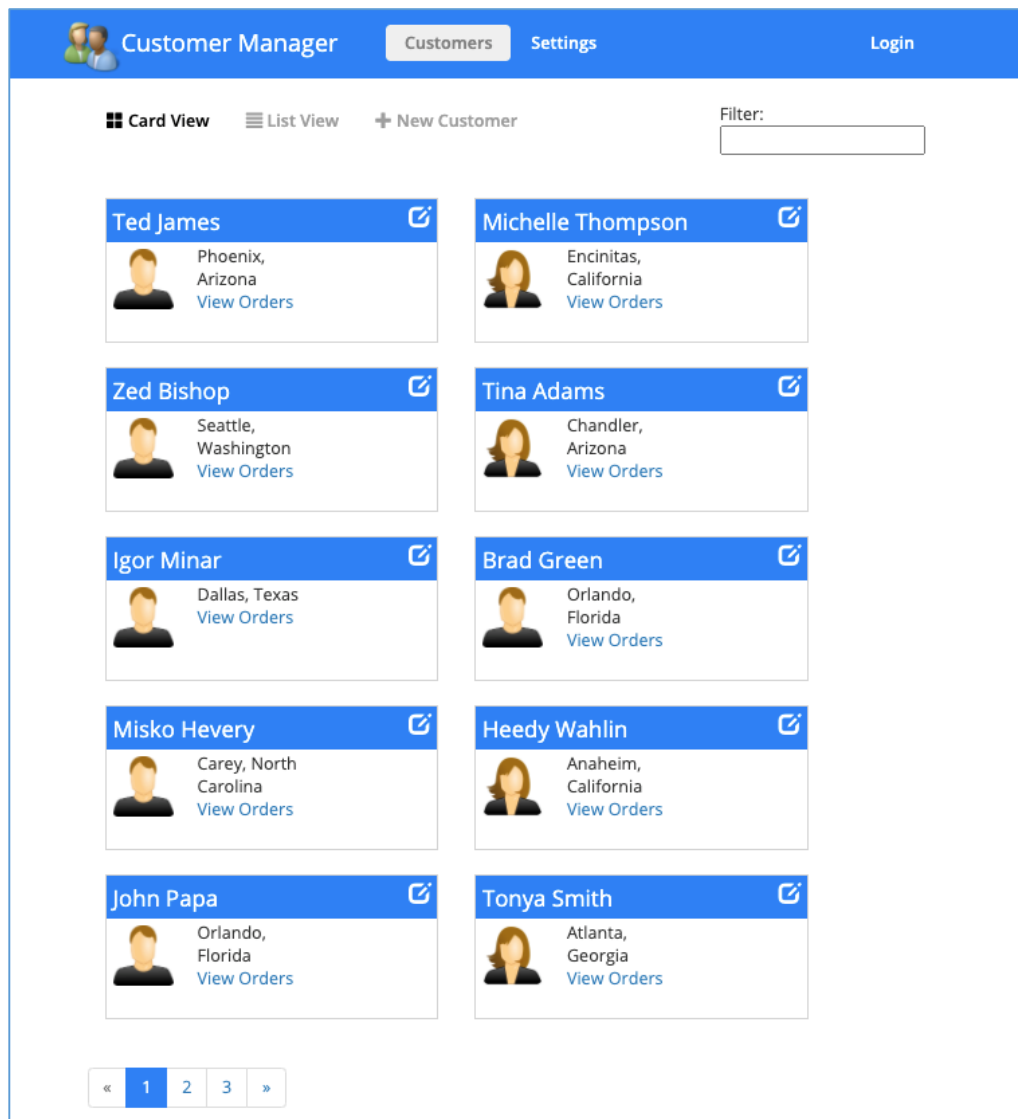
- Home page (URL: /)
 - Redirect to /customers
- Customers page (URL: /customers)
 - Display Card View (with Filter and pagination functionalities)
 - Display List View (with Filter and pagination functionalities)
 - New Customer (Must logged in to add new customer if not redirect to /login)
- Customer detail page (URL: /customers/:id/details)
 - Display Customer in detail
 - Back to View all Customers
- Customer orders page (URL: /customers/:id/orders)
 - Display Orders for current Customer
 - Back to View all Customers
- Customer edit page (URL: /customers/:id/edit)
 - Edit Customer's information such as: First Name, Last Name, Address, City, State
 - Delete Customer
 - Back to View all Customers

Styles


Use Bootstrap 4 for styling.

Layout must match UI design. Color, icons, fonts,... does't need to match but make sure your web is clean and easy to follow.

Responsive design is not required.

Screen Design:**Customers (Card View):**

Customers (List View):

 Customer Manager

Customers Settings











Login

Card View

List View

New Customer

Filter:

	First Name	Last Name	Address	City	State	Order Total	
	Ted	James	1234 Anywhere St.	Phoenix	Arizona	\$207.98	View Orders
	Michelle	Thompson	345 Cedar Point Ave.	Encinitas	California	\$8.98	View Orders
	Zed	Bishop	1822 Long Bay Dr.	Seattle	Washington	\$229.97	View Orders
	Tina	Adams	79455 Pinetop Way	Chandler	Arizona	\$25.48	View Orders
	Igor	Minar	576 Crescent Blvd.	Dallas	Texas	\$469.98	View Orders
	Brad	Green	9874 Center St.	Orlando	Florida	\$29.98	View Orders
	Misko	Hevery	9812 Builtway Appt #1	Carey	North Carolina	\$321.97	View Orders
	Heedy	Wahlin	4651 Tuvo St.	Anaheim	California	\$679.98	View Orders
	John	Papa	66 Ray St.	Orlando	Florida	\$29.98	View Orders
	Tonya	Smith	1455 Chandler Blvd.	Atlanta	Georgia	\$313.97	View Orders

«


1


2




3

»

New Customer (Must login)

 Customer Manager Customers Settings test@test.com (Logout)

 Customer Information

 Customer Details  Customer Orders  New Customer

First Name

Last Name

Address


City


State




Cancel


Insert

[View all Customers](#)**Customer Detail:**

 Customer Manager Customers Settings test@test.com (Logout)

 Customer Information

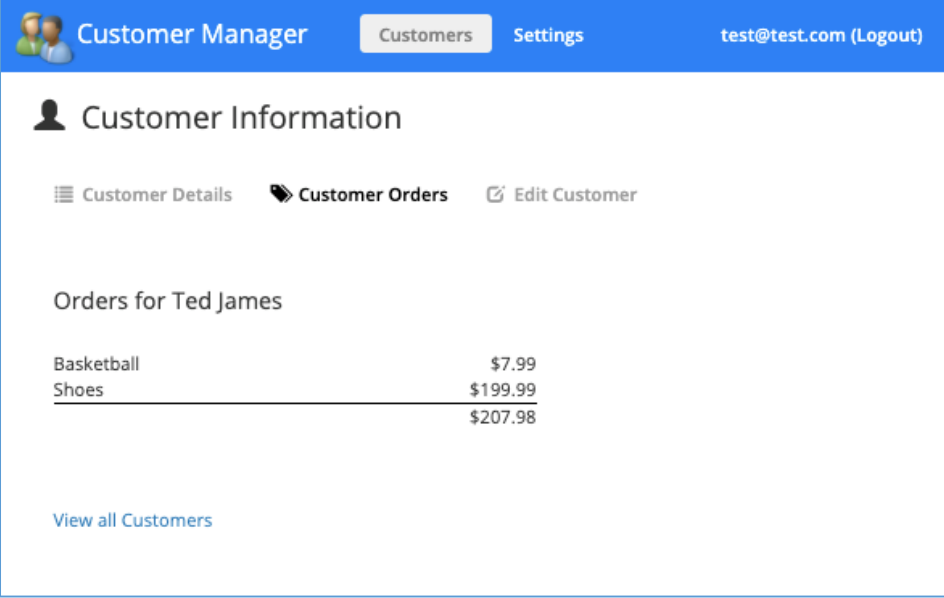
 Customer Details  Customer Orders  Edit Customer



Ted James

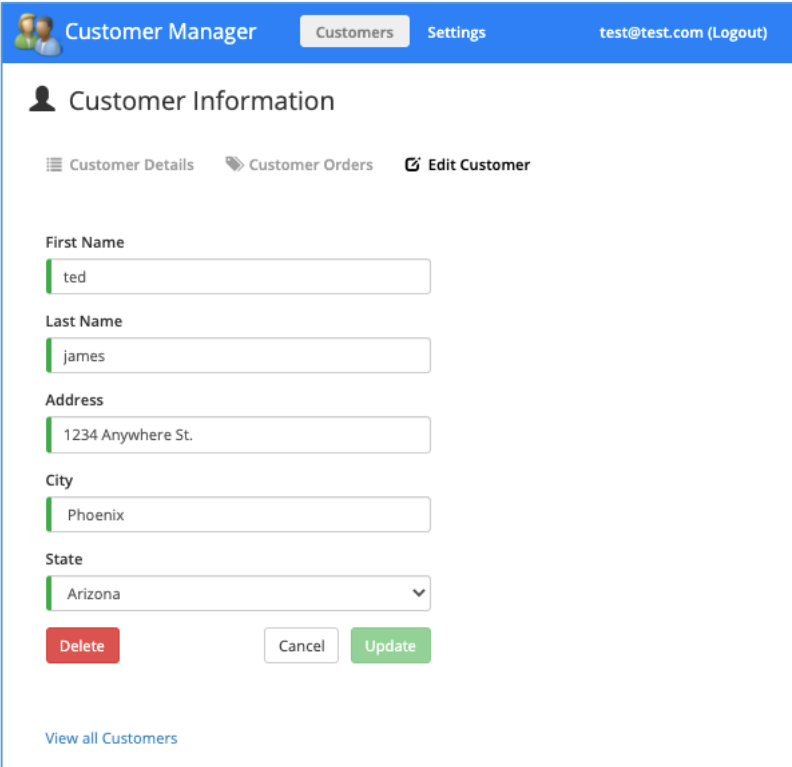
1234 Anywhere St.
Phoenix , Arizona

[View all Customers](#)

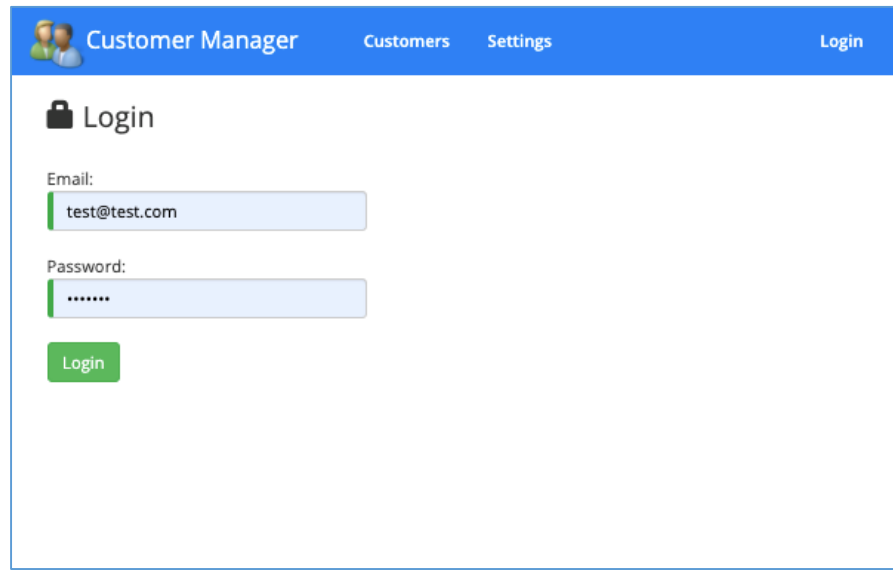
Customer Orders:

The screenshot shows the 'Customer Manager' application. The top navigation bar is blue with the 'Customer Manager' logo, 'Customers' and 'Settings' tabs, and a user profile 'test@test.com (Logout)'. The main content area is titled 'Customer Information' and has three tabs: 'Customer Details', 'Customer Orders' (which is active), and 'Edit Customer'. Below the tabs, the section is titled 'Orders for Ted James'. It displays a table with two rows: 'Basketball' for \$7.99 and 'Shoes' for \$199.99. A horizontal line separates the two rows, and the total amount, \$207.98, is shown to the right. At the bottom left, there is a link 'View all Customers'.

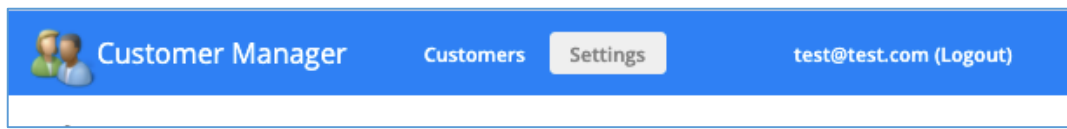
Basketball	\$7.99
Shoes	\$199.99
<hr/>	
	\$207.98

Edit Customer:

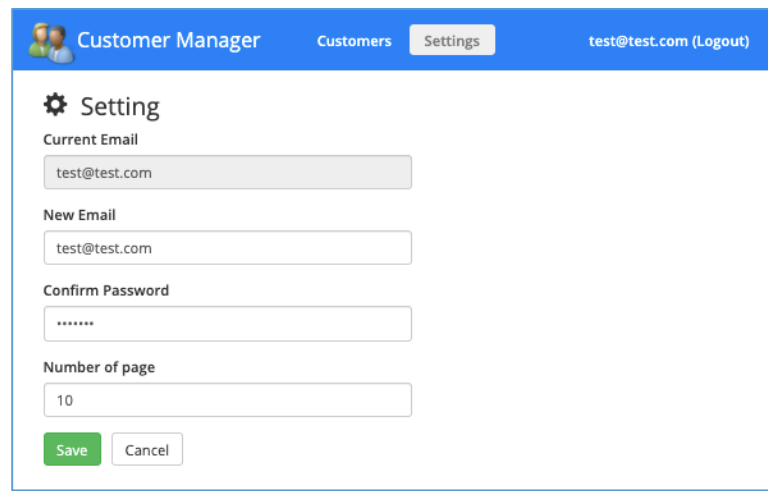
The screenshot shows the 'Customer Manager' application with the 'Edit Customer' tab active. The form contains the following fields: 'First Name' (ted), 'Last Name' (james), 'Address' (1234 Anywhere St.), 'City' (Phoenix), and 'State' (Arizona). At the bottom, there are three buttons: 'Delete' (red), 'Cancel' (white), and 'Update' (green). A link 'View all Customers' is at the bottom left.

Login:

The screenshot shows the 'Customer Manager' application interface. The top navigation bar is blue and contains the application name 'Customer Manager' with a user icon, and two links: 'Customers' and 'Settings'. On the far right of the bar is a 'Login' link. Below the navigation bar, the main content area is titled 'Login' with a lock icon. It contains two input fields: 'Email:' with the value 'test@test.com' and 'Password:' with masked characters '*****'. A green 'Login' button is positioned below the password field.

Logout:

The screenshot shows the 'Customer Manager' application interface after a successful login. The top navigation bar is blue and contains the application name 'Customer Manager' with a user icon, and two links: 'Customers' and 'Settings'. On the far right of the bar, the user's email 'test@test.com' is displayed next to a '(Logout)' link. Below the navigation bar, the main content area is empty.

Settings (Optional):

The screenshot shows the 'Customer Manager' application interface with the 'Settings' page selected. The top navigation bar is blue and contains the application name 'Customer Manager' with a user icon, and two links: 'Customers' and 'Settings'. On the far right of the bar, the user's email 'test@test.com' is displayed next to a '(Logout)' link. Below the navigation bar, the main content area is titled 'Setting' with a gear icon. It contains four input fields: 'Current Email' with the value 'test@test.com', 'New Email' with the value 'test@test.com', 'Confirm Password' with masked characters '*****', and 'Number of page' with the value '10'. At the bottom of the form are two buttons: a green 'Save' button and a white 'Cancel' button.

Backend Specs

You are given a project inside customer-management-api.zip.

Unzip, install dependencies (`npm install`) and start the project (with `npm start`). You should see the output: `Express listening on port 3000`.

BASE URL: <http://localhost:3000/>

Endpoints:

Login:

1. POST /api/auth/login

Example request body:

```
1. {  
2.   "email": "test@test.com",  
3.   "password": "test123"  
4. }
```

No authentication required, returns a User

Required fields: email, password

Logout:

1. POST /api/auth/logout

Example request body:

```
1. {  
2. }
```

Authentication required, returns boolean (true/false)

Update User (Settings page)

1. POST /api/users

Example request body:

```
1. {  
2.   "currentEmail": "test@test.com",  
3.   "newEmail": "test123@test.com",  
4.   "password": "test123",  
5.   "numberPerPage": 10  
6. }
```

Authentication required, returns the updated User

Required fields: currentEmail, newEmail, password

List All Customers

1. GET /api/customers

No authentication required, returns array of Customer

List All Customers by page

2. GET /api/customers/page/:skip/:top

No authentication required, returns array of paginated Customer where skip is how many element to skipped and top is the number of item to return.

Example 1: GET /api/customers/page/0/10

Would return that first 10 Customers

Example 2: GET /api/customers/page/10/10

Would skip first 10 Customers and return next 10 Customers

List Customer by Id

1. GET /api/customers/:id

Returns Customer by Id

No authentication required, will return single Customer or Error if not found any Customer

Edit Customer By Id

```
1. PUT /customers/:id
```

Example request body:

```
2. {
3.   "firstName": "Teddy",
4.   "lastName": "James",
5.   "Address": "123 New Addres",
6.   "city": "city",
7. }
```

Authentication required, returns the updated Customer

Required fields: firstName, lastName, address, city

Delete Customer By Id

```
1. DELETE /customers/:id
```

Authentication required, returns the deleted Customer

JSON Objects returned by API:

Make sure the right content type like Content-Type: application/json; charset=utf-8 is correctly returned.

Users (for authentication)

```
1. {
2.   "id": 1,
3.   "email": "jake@jake.jake",
4.   "password": "jwt.token.here",
5.   "numberPerPage": 10
6. }
```

Customers

```
1. [{
2.   "id":1,
3.   "firstName":"ted111",
4.   "lastName":"james",
5.   "gender":"male",
6.   "address":"1234 Anywhere St.",
7.   "city":" Phoenix",
8.   "state":{"abbreviation":"AZ","name":"Arizona"},
9.   "orders":[
10.    {"productName":"Basketball","itemCost":7.99},
11.    {"productName":"Shoes","itemCost":199.99}],
12.   "latitude":33.299,"longitude":-111.963,
13.   "orderTotal":207.98000000000002
14. },...]
```

Customers By page

```
1. {
2.   "results": [{
3.     "id":1,
4.     "firstName":"ted111",
5.     "lastName":"james",
6.     "gender":"male",
7.     "address":"1234 Anywhere St.",
8.     "city":" Phoenix",
9.     "state":{"abbreviation":"AZ","name":"Arizona"},
10.    "orders":[
```

```
11.      {"productName":"Basketball","itemCost":7.99},
12.      {"productName":"Shoes","itemCost":199.99}],
13.      "latitude":33.299,"longitude":-111.963,
14.      "orderTotal":207.98000000000002
15.    },...],
16.    "totalRecords": 22
17. }
```

Errors and Status Codes

If a request fails any validations, expect a 400 and errors in the following format:

```
1. {
2.   "message": "Input is invalid"
3. }
```

Other status codes:

401 for Unauthorized requests, when a request requires authentication but it isn't provided

403 for Forbidden requests, when a request may be valid but the user doesn't have permissions to perform the action

404 for Not found requests, when a resource can't be found to fulfill the request.