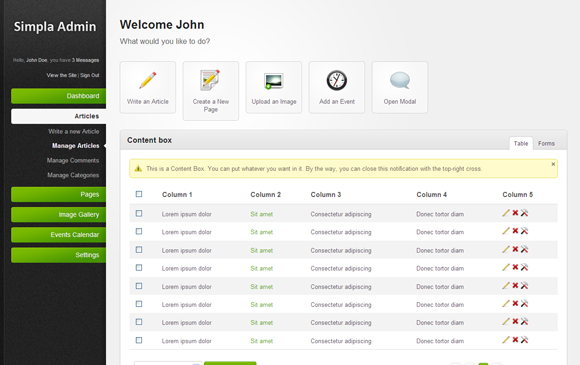
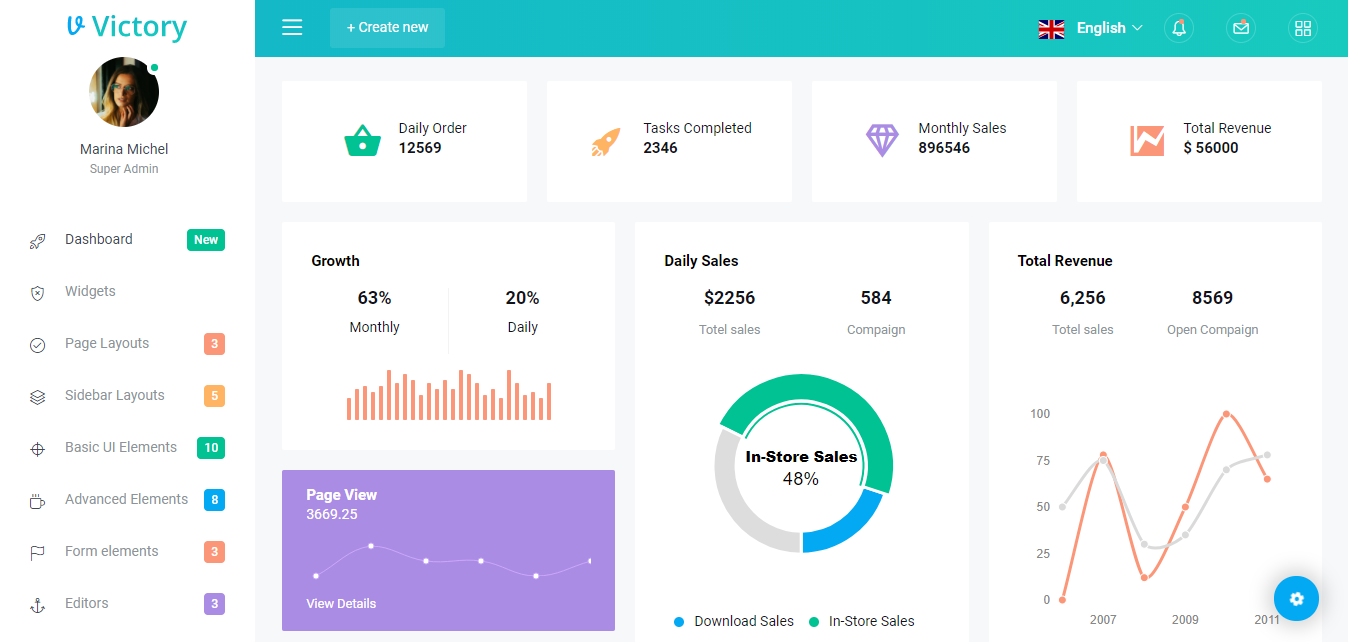
**Couse Project**

**Project Requirements:**

1. Complete the following in your E-commerce project, and on the backend API you did in NodeJS project [You can use any backend APIs (NodeJS, or .Net)]:
   1. **User part (E-commerce, and complete on the assignments)**:
      * Add home page that displays some info about your E-commerce.
      * All components except for home component must be lazy loaded.
      * Create products module, and assign all components related to products on it.
      * In products page, display the products in Cards (like: souq.com)
      * When user clicks on product image, it opens in a pop-up modal window (Use Angular material)
      * In the products table add text box for entering the needed quantity, and ‘Add to cart’ button that adds the selected item with the entered wanted quantity to the shopping cart.
      * In the page header add shopping cart icon that displays the number of added items, and when click it opens a new page with ordered products details (Use lazy loading), and to complete the purchase process. [Like souq.com].
      * Handle that when user buys an item, update the item quantity in real-time for the other connected users.
   2. **Admin Part:** [Use new separate project and hidden URL for admin pages]:
      * Handle authentication for admins:
        + Add table for admins and their username and passwords.
        + Add login page for admin.
        + Use routes guards, to handle that admin pages can’t be reached without login.
        + Handle session time, and logout.
        + Use lazy loading for all admin pages.
      * Basic admin functionalities:
        + List all products in Data table (Use Angular material), and table should have: details, edit, delete buttons.
        + Details button: will open the product details in modal pop-up window (Use Angular material or NGX-bootstrap).
        + Update products: On the products table add ‘Edit’ button that opens the products details in another page (or popup window) with editing enabled, and ‘Cancel’, ‘Update’ buttons.
        + Delete: On the products table add ‘Delete’ button, and when user clicks it, show confirmation message before deleting.
        + Your admin panel should be like the following [You can use: “Booststrap Lite admin” template]



* 1. **Project Bonus**:
* Make a dashboard in the admin panel that displays some reports about: number of sells, best sell products, sells per day/month/year …, like the following:



* + Search on NPM for Angular charts components.
* Handle localization and internationalization for your application (Arabic and English).
* Use PayPal integration to complete the purchase process.
* In the admin panel, add pages: Add new admin, view/ edit/ delete admins.
* In admin panel, add a page where admin can update his information, and change his password.
* Handle Forget password for admin, and users.
* Handle register, login and authentication for users.

**Self-Study (Enhance your Angular knowledge):**

1. Read the following topics in Angular documentations, and apply them in a demo:
   1. More about components & Templates:
      * <https://angular.io/guide/component-styles>
      * <https://angular.io/guide/elements>
      * <https://angular.io/guide/dynamic-component-loader>
      * <https://angular.io/guide/animations>
   2. Bootstrapping: <https://angular.io/guide/bootstrapping>
   3. Angular NgModules:
      * <https://angular.io/guide/ngmodules>
      * <https://angular.io/guide/ngmodule-vs-jsmodule>
      * <https://angular.io/guide/frequent-ngmodules>
      * <https://angular.io/guide/module-types>
      * <https://angular.io/guide/entry-components>
      * <https://angular.io/guide/feature-modules>
      * <https://angular.io/guide/providers>
      * <https://angular.io/guide/singleton-services>
      * <https://angular.io/guide/lazy-loading-ngmodules>
      * <https://angular.io/guide/sharing-ngmodules>
      * <https://angular.io/guide/ngmodule-api>
   4. More about Dependency Injection in Angular:
      * <https://angular.io/guide/dependency-injection-pattern>
      * <https://angular.io/guide/dependency-injection>
      * <https://angular.io/guide/hierarchical-dependency-injection>
      * <https://angular.io/guide/dependency-injection-in-action>
   5. More about Observables, RxJS:
      * <https://angular.io/guide/observables>
      * <https://angular.io/guide/rx-library>
      * <https://angular.io/guide/observables-in-angular>
      * <https://angular.io/guide/practical-observable-usage>
      * <https://angular.io/guide/comparing-observables>
   6. Unit tests for angular apps: <https://angular.io/guide/testing>
   7. Angular Internationalization: <https://angular.io/guide/i18n>
   8. Angular Language Service: <https://angular.io/guide/language-service>
   9. Angular service worker: <https://angular.io/guide/service-worker-intro>
   10. Subjects: [https://stackoverflow.com/questions/34376854/delegation-eventemitter-or-observable-in-angular/35568924#35568924](https://stackoverflow.com/questions/34376854/delegation-eventemitter-or-observable-in-angular/35568924)
   11. Security guidelines: <https://angular.io/guide/security>
   12. Angular cheat sheet: <https://angular.io/guide/cheatsheet>