ArialShop Documentation - Vue2.0

Installation Instructions (Single page progressive e-commerce app using VueJS)

Softwares

Make sure you have **Node** version >= 8.0, **NPM** >= 5 and **MongoDB**

- Download and unzip the file from **codecanyon**
- navigate inside the directory

```
cd arialshop
```

Start database

- Start mongodb in a separate shell

Try-1: `npm run database`

Try-2: In Windows operating system we can start it by opening the following file

```
C:/Program Files/MongoDB/Server/3.2/bin/mongod.exe
```

Start development server

- Open another terminal (command prompt)

1. install the dependencies with npm
   `npm i`

2. serve with hot reload at localhost:8080
   `npm run dev`

- build for production with minification
  `npm run prod`

- build client for production and view the bundle analyzer report
  `npm run build --report`

Deploy to Heroku

1. Run command `npm run prod`

2. Install mLab heroku addon

```
heroku addons:create mongolab
```

3. Upload everything inside dist directory to Heroku

4. Run command `npm i --production`

5. Check heroku logs `heroku logs --tail` for any error

Client Settings

Path: `src/config.js`

Language Settings

```
export const lang = 'en' // Change to 'fr' for french version.
```

The language file is inside the `src/i18n` directory (fr.js)

Timeout settings

```
export const typingTimeout = 500 // Used for searching. e.g. search.vue
export const loadingTimeout = 200 // Used for showing the loading indicator
```

Clear Cart Settings

```
export const clearCart = false // Whether to clear the cart after order is
```

Enable / Disable demo mode

Restricts users from saving data if enabled

```
export const demo = false
```

User Roles

```
export const userRoles = ['user', 'vendor', 'manager', 'admin'] // This sho
```

Review Settings

```
export const reviewSettings = {
    enabled: true, // Enables review for products
    moderate: false // If enabled, the review will be visible to public aft
}
```

List of order and payment status for Order Management

```
export const orderStatuses = ['Payment Pending', 'Order Placed', 'Order Accept
export const paymentStatuses = ['Pending', 'Cancelled', 'Paid']
```

Enabled Payment Methods for ecommerce

```
export const paymentMethods = ['PayPal', 'COD']
```

Regional Settings

```
export const country = { name: 'India', code: 'IN' }
export const currency = {
    symbol: '₹',
    code: 'INR', // Shop currency
    paypal: 'USD',// Paypal currency code *** Please choose from https://devel
    exchange_rate: '0.015' // Paypal currency code(USD) / Shop currency (INR)
}
```

Menu for Dashboad and Header

```
export const menuItems = [
  { name: 'Products', url: '/admin/products', icon: 'store', authenticate: 've
  { name: 'My Orders', url: '/admin/orders', icon: 'watch_later', authenticate
  { name: 'Manage Orders', url: '/admin/orders/manage', icon: 'history', authe
  { name: 'Address', url: '/admin/address', icon: 'location_city', authenticat
  { name: 'Manage Reviews', url: '/admin/reviews/manage', icon: 'stars', authe
  { name: 'My Reviews', url: '/admin/reviews', icon: 'star_rate', authenticate
  { name: 'My Wishlist', url: '/admin/wishlist', icon: 'favorite', authenticat
  { name: 'Brands', url: '/admin/brands', icon: 'wb_auto', authenticate: 'mana
  { name: 'Categories', url: '/admin/categories', icon: 'view_comfy', authenti
  { name: 'Features', url: '/admin/features', icon: 'check_circle', authentica
  { name: 'Coupons', url: '/admin/coupons', icon: 'style', authenticate: 'mana
  { name: 'Shippings', url: '/admin/shippings', icon: 'local_shipping', authen
  { name: 'Users', url: '/admin/users', icon: 'face', img: 'auth.png', authent
  { name: 'Media Library', url: '/admin/media', authenticate: 'user', icon: 'p
  { name: 'Profile', url: '/account/profile', authenticate: 'user', icon: 'fac
  { name: 'Change Password', url: '/account/change-password', authenticate: 'u
]
```

Environment Settings

Change `.env` file according to your credentials
Generate the following APIs and input the credentials into .env file present at the root of the project

Email Client

Sendgrid

Payment Gateway

Paypal Developer

Login through Facebook

Facebook Developer

Google Auth Login

Google Developer Console

Use twitter account to login to ArialShop

Twitter Developer

```
MONGODB_URI=mongodb://localhost:27017/arialshop-dev
DOMAIN=http://localhost:8080
SESSION_SECRET=arialshop-secret
PAYPAL_MODE=sandbox
PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
PAYPAL_CLIENT_SECRET=YOUR_PAYPAL_CLIENT_SECRET
STRIPE_APIKEY=sk_test_REST_OF_YOUR_KEY
SENDGRID_APIKEY=YOUR_SENDGRID_API_KEY
FACEBOOK_ID=YOUR_FACEBOOK_ID
FACEBOOK_SECRET=YOUR_FACEBOOK_SECRET
TWITTER_ID=YOUR_TWITTER_ID
TWITTER_SECRET=YOUR_TWITTER_SECRET
GOOGLE_ID=YOUR_GOOGLE_ID
GOOGLE_SECRET=YOUR_GOOGLE_SECRET
GOOGLE_MAPS_API=YOUR_GOOGLE_MAPS_API
```

Project Root Directory Structure

```
|   .babelrc
|   .dockerignore
|   .editorconfig
|   .env
|   .gitignore
|   .postcssrc.js
|   copy-dist-files.js
|   Dockerfile
|   documentation.pdf
|   index.html
|   out.txt
|   package-lock.json
|   package.json
|   Procfile
|   README.md
|   tsconfig.json
|   tslint.json
|
+---build
+---config
+---server
+---src
+---static
```

Client Directory Structure

```
+---src
|   |   App.vue
|   |   config.js
|   |   main.js
|   |
|   +---assets
|   |       .gitkeep
|   |
|   +---components
|   |   |
|   |   \---modal
|   |
|   +---filters
|   |       index.js
|   |
|   +---i18n
|   |       fr.js
|   |
|   +---mixins
|   |       basic.js
|   |
|   +---pages
|   |   |
|   |   +---account
|   |   |
|   |   +---admin
|   |   |   |   index.vue
|   |   |   |
|   |   |   +---address
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   +---brands
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   +---categories
|   |   |   |       index.vue
|   |   |   |
|   |   |   +---coupons
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   +---features
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   +---media
|   |   |   |       index.vue
|   |   |   |
|   |   |   +---orders
|   |   |   |       index.vue
|   |   |   |       manage.vue
|   |   |   |
|   |   |   +---products
|   |   |   |       index.vue
|   |   |   |
|   |   |   +---reviews
|   |   |   |       index.vue
|   |   |   |       manage.vue
|   |   |   |
|   |   |   +---shippings
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   +---users
|   |   |   |       index.vue
|   |   |   |       _id.vue
|   |   |   |
|   |   |   \---wishlist
|   |   |           index.vue
|   |   |
|   |   +---auth
|   |   |
|   |   \---_slug
|   |           _id.vue
|   |
|   +---router
|   |       index.js
|   |
|   \---store
|
\---static
    |   manifest.json
    |   product-placeholder.png
    +---icons
    \---uploads
        \---thumb
```

Server Directory Structure

```
+---server
|   |   app.ts
|   |   config.ts
|   |   routes.ts
|   |   seed.ts
|   |   tsconfig.json
|   |
|   +---api
|   |   |   base.ts
|   |   +---address
|   |   +---brand
|   |   +---category
|   |   +---contact
|   |   +---coupon
|   |   +---customer
|   |   +---feature
|   |   +---media
|   |   +---order
|   |   +---orderHistory
|   |   +---pay
|   |   +---payment-method
|   |   +---product
|   |   +---review
|   |   +---sendmail
|   |   +---shipping
|   |   +---upload
|   |   +---user
|   |   \---wishlist
|   |
|   +---auth
|   |   |   auth.service.ts
|   |   |   index.ts
|   |   |
|   |   +---facebook
|   |   |
|   |   +---google
|   |   |
|   |   +---local
|   |   |
|   |   \---twitter
|   |
|
```

Usage Instructions

Home

This is the main page of our VueJS progressive e-commerce app. Here we get

- List of all products
- Filter Products: based on `Price (Price Slider), Brand, Features (Color, Type, Fit, Fabric, Neck)`
- Sort: Based on Price and Name

Each product contains a add to cart button. Once the product is added into the cart, we get the increase or decrease cart
quantity option By clicking each product we arrive at the product detail page

Product Details

This page presents the complete details of the product

- Product name
- Description
- Price, MRP
- Product Image (Including additional images)
- Brand
- Category
- Quantity in cart
- Size
- Features

Search

The top navigation bar of the website has a search box which autocompletes with product info while user
starts typing. By clicking a suggested item in the search bar, the page navigates to the product details page of the selected
product.

Category

Get the current category name with all products under it
This page also has all the filter and sort options.

Shopping Cart

This store is featured with a shopping cart facility which is easy to use and fast.

- Get quick summary of what is there in Cart
- Modify the cart quantity
- Checkout using Paypal

Login / Signup

Features like Signup / SignIn / Change Password / Logout is integrated into this application already with high level of security,
so that you no longer need to be worry about implementing all those features into the application

A user need **not** have to navigate to a separate page to login or signup. It comes as a popup which is a huge ui improvement.
This login popup has a advantage of poping out for any route when a guest user tries to access a restricted page Both the
login and signup page has the option for connect using facebook, twitter, google as well

Checkout

- The checkout page Displays the Order Amount + ShipCharge
- This also has an option discount coupons which is appif valid.
- Here the user can choose the Payment options (CasDelivery, Paypal)
- This page automatically choose the best available ShipOptions based on the total order weight and the shipper availability.
- While checkout the user can choose from any saddress.

Address Management

The address management is integrated into the checkout page to make the checkout experience single view
and easy. Here the user can store and manage different addresses.

Order Management

The user has the facility to view the order history. Administrators can change order status + payment status
Users: All the orders placed by the logged in user is available in this view.
Administrators: This view presents all orders placed by users with the option to change order status and shipping

User Management

Users:
Change Password
Forgot Password

Administrators:
User role management

Media Library

- Now the shop has a new media library where the managers can upload any image that is to be used in the shopping application
- Clicking on each image displays the details about iwell as an option to delete it

Products (Role: Managers, Administrators)

Product details can be added, modified and deleted using this page. Each product can be associated into
a single Brand, Category A product can have

- Multiple features
- Multiple key features
- Multiple product images
- The list contains all the available products with a sebox to filter the list.
- Clicking on a product at the product list will poputhe details of the product at the right sidebar
- The right sidebar has option to change product ndetails, brand, category
- This sidebar also contains a module to manage proimages

Manage Brands

Administrators can add, edit, delete, filter brands of their store from this view

Manage Categories

- Categories are presented in Parent-Child manner in store for better organisation of products.
- Store’s navigation bar at top contains all the categoarranged in parent-child fashion.
- This view provides facility to add both parent and ccategories, re-arrange category association according to their
requirement.

Server Settings

Path: *server/config.ts*

Website Settings

```
emailEnabled = true;
```

Review Settings

```
reviewSettings = {
  enabled: true, // Enables review for products
  moderate: false // If enabled, the review will be visible to public after ad
};
```

Product Settings

```
product = { moderate: false };
```

User Roles

```
userRoles = ['user', 'vendor', 'manager', 'admin']; // This should be in ascen
```

Forgot Password Email Settings

```
forgotPasswordEmail = (body) => { // Expects email id and password reset token
  return {
    from: 'passwordreset@codenx.com',
    to: body.email,
    subject: 'ArialShop Password Reset Request',
    text: 'You are receiving this because you (or someone else) have requested
    'Please click on the following link, or paste this into your browser to co
    process.env.DOMAIN + '/account/reset-password/' + body.token + '\n\n' +
    'If you did not request this, please ignore this email and your password w
  }
}
```

Reset Password Email Settings

```
resetPasswordEmail = (body) => { // Expects email id and name
  return {
    from: 'passwordreset@codenx.com',
    to: body.email,
    subject: 'ArialShop Password Changed',
    text: 'Hello,\n\n' +
    'This is a confirmation that the password for your account ' + body.to + '
  };
}
```

Order Placed Email Settings

```
orderPlacedEmail = (body) => { // Expects email id, orderNo, ...
  return {
    from: 'CodeNx <admin@codenx.com>',
    to: body.to,
    subject: 'Order Placed Successfully',
    text: 'Order No: ' + body.orderNo
    + '\n Status: ' + body.status
    + '\n\n Payment Method: ' + body.payment_method
    + '\n\n Payment ID: ' + body.id
    + '\n Amount: ' + body.amount.currency + ' ' + Math.round(body.amount.tota
    + '\n\n Name: ' + body.address.recipient_name
    + '\n Address: ' + body.address.line1
    + '\n City: ' + body.address.city
    + '\n Zip: ' + body.address.postal_code
  };
}
```

Order Updated Email Settings

```
orderUpdatedEmail = (body) => {
  return {
    from: 'CodeNx <admin@codenx.com>',
    to: body.to,
    subject: 'Your Order Status Updated',
    text: 'Order No: ' + body.orderNo
    + '\n Status: ' + body.status
    + '\n\n Payment Method: ' + body.payment_method
    + '\n\n Payment ID: ' + body.id
    + '\n Amount: ' + body.amount.currency + ' ' + Math.round(body.amount.tota
    + '\n\n \n Name: ' + body.address.recipient_name
    + '\n Address: ' + body.address.line1
    + '\n City: ' + body.address.city
    + '\n State: ' + body.address.state
    + '\n Zip: ' + body.address.postal_code
  };
}
```