

Proposal

e-Commerce website: Vidia

From: SteveTops Team (Team 2)

To: Steve George

Date: 29 November 2020

1. Description:

If you would like to experience the best online portal to buy a laptop, you are in the right place. Vidia is the ultimate destination for your brilliant mind's processing needs. We are the host to a wide array of all the major manufacturers. It is time to redefine your companion. By this, we mean your friendly laptop with our treasure-trove of trendy laptops. Our online store brings you the latest models straight out of the production house and that too at the best market rates. We guarantee that you won't be able to find such mesmerizing deals anywhere else. You can shop online at vidia.uwpace.ca from the comfort of your couch and get your device delivered to your doorstep at no delivery cost within Canada.

2. Project Details:

At the Vidia e-Commerce website, we have laptops to sell as our single product selling website, based on your requirement. You can filter your search by selecting categories based on available brands, processors, and graphics cards. The following views and their functionalities are proposed below for users and admin users under the frontend and backend of the website.

- **Frontend -**

Anyone visiting the Vidia e-Commerce website will have:

- **Home** page of the Vidia e-Commerce website and links to visit the website from there.
- **List view** page of available laptops with the categories to filter the list view under brands, processors, and graphics cards.
- **Detail view** page of a selected laptop from list view to know the details of it, including a feature to add it to the shopping cart.
- **Shopping cart** page where user can see the items added to the cart and user will have the option to edit the quantity or to delete the item from the cart.
- **Checkout** page to view the final items selected, add shipping date, choose from available addresses or add a new address, and payment details panel to add card details to go for payment and finalize the order.**
- **Successfully placed order** page after the payment is done successfully via payment gateway and redirected to this page.**

- **Your Orders** page where the user can see the placed orders under their login. **
- **Login** page to log into the website for Checkout page along with Successfully placed order page, and Your Orders pages to visit.
- **Register** page for a new user to register into the website to place an order with Vidia.
- **Profile** page where users can see their details already available that they entered while registering with the website. Other details to add like contact number, address, image, first name, last name, and option to delete the account.**
- Feature to **logout** from the logged-in status.**
- **Social Media Links** to visit the Facebook page, Insta page, and Twitter page.
- **Privacy Policy** page to display policies undertaken for the Vidia e-Commerce website.
- **Copyright** page to display the copyright details of the website for user information.
- **FAQ** page where frequently asked questions will be posted for users to have a look.
- **About us** page to display the information about the company - Vidia e-Commerce
- **Contact us** page will be a form for a user to contact with their name, email address and query. Also, contact details like link on Google maps, contact number and mailing address will be present there.

**** - users need to be logged in to view these pages/features.**

- **Backend -**

Backend views are the pages for admin users to view once they login into the website to add, update or delete the records from the database tables of the website. We are following soft deletes on our database, so if the admin deletes the record, the data will be available in the table as history data.

Also, as we follow the soft delete on the database, if the admin deletes the record, the list view on the respective pages can still display the record as an inactive record without the link to delete it. In Vidia e-Commerce, we are maintaining 11 tables for the project. Hence in Backend view, we will have the following pages for Admin:

- **Dashboard** page for the admin to have the evaluated data based on the data available with the database and the data captured from the number of users registered, and the orders placed.
- **Users** page as a list view for admin. A link to **Add New User** page to add new users, a link to **Edit User** page to update user details, and a link as a feature to **delete** users of the website.
- **Laptops** page as a list view for admin. A link to the **Add New Laptop** page to add a new laptop, a link to the **Edit Laptop** page to update

- laptop details, and a link as a feature to **delete** existing laptops to display on the frontend website.
- **Brands** page as a list view for admin. A link to **Add New Brand** page to add a new brand, a link to **Edit Brand** page to update brand details, and a link as a feature to **delete** existing brands to display on the frontend website.
 - **Processors** page as a list view for admin. A link to **Add New Processor** page to add a new processor, a link to **Edit Processor** page to update processor details, and a link as a feature to **delete** existing processors to display on the frontend website.
 - **Graphics Cards** page as a list view for admin. A link to **Add New Graphics Card** page to add a new graphics card, a link to **Edit Graphics Card** page to update graphics card details, and a link as a feature to **delete** existing graphics cards to display on the frontend website.
 - **Addresses** page as a list view for admin. A link to the **Add New Address** page to add a new address, a link to the **Edit Address** page to update address details, and a link as a feature to **delete** existing addresses.
 - **Provinces** page as a list view for admin. A link to the **Add New Province** page to add a new province, a link to the **Edit Province** page to update province details, and a link as a feature to **delete** existing provinces.
 - **Comments** page as a list view for admin. A link to the **Add New Comment** page to add a new comment, a link to the **Edit Comment** page to update comment details, and a link as a feature to **delete** existing comments.
 - **Orders** page as a list view for admin. A link to **Add New Order** page to add a new order with the option to select laptops from laptops list, the quantity for the selected laptop, add address details, shipping date, and payment details to make the payment and finalize the order, a link to **Edit Address** page to update order and its order details, and a link as a feature to **delete** existing orders.

3. Proposed Design Solution:

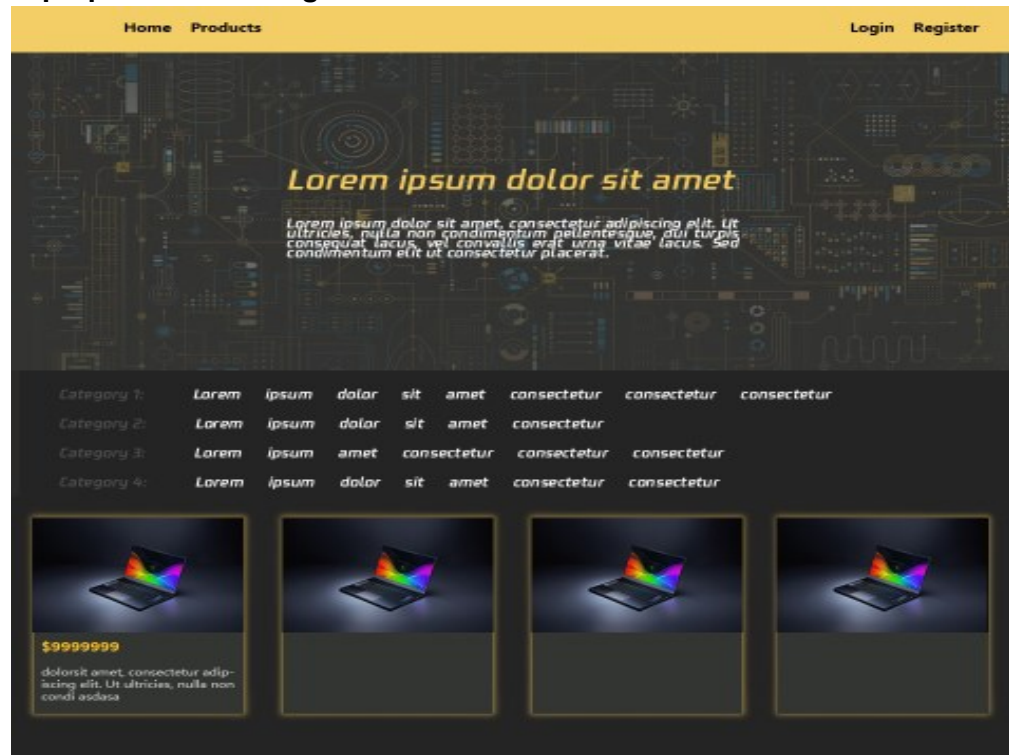
Pictures below provides proposed graphical solutions with all the features and functionality of the project. With the various research work done on the look and feel as per the latest running trends on the e-commerce websites, and the expert designing skills of our designer, we created three pages to submit as the proposed designs.

The design is done by creating PSDs on Adobe Photoshop. The HTML pages are designed to implement those PSDs by slicing them into necessary images and using expert CSS skills. The screenshots of the three pages viz Home page, Laptops List view page and Laptop's detail view page are taken from the PSDs created and are displayed here to see the preliminary view of our proposed designs.

- **Home Page:**



- **Laptops List View Page:**



- **Laptop's Detail View Page:**



4. Use Cases:

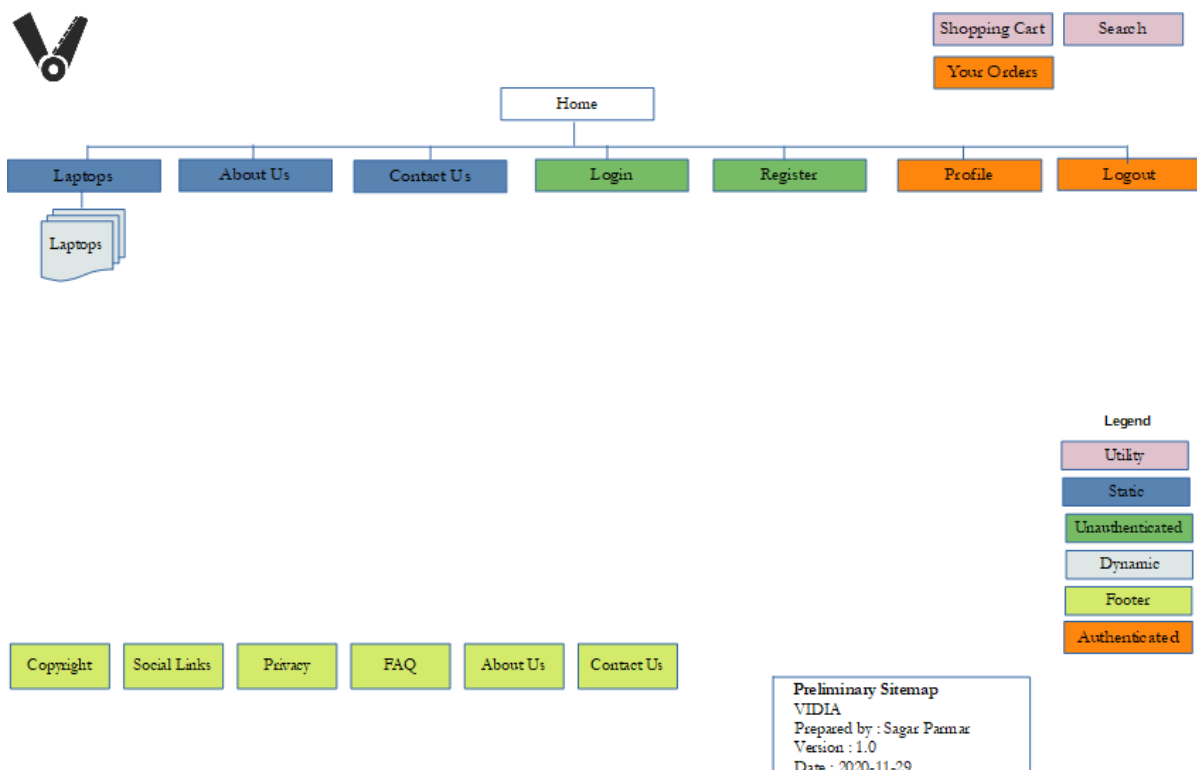
Guest: Guest Users would be able to surf all pages except the profile page. This user does not require to log in. They could add items to the cart that they are interested in, but they won't be able to order it until they sign up or log in.

Authenticated User: Authenticated users would be able to explore all of the site's pages, including their profile page. They can also put items in the cart and order them whenever they are ready. While ordering, they would have a chance of choosing their preferred shipping address, and if they haven't added any address, then they can add it too.

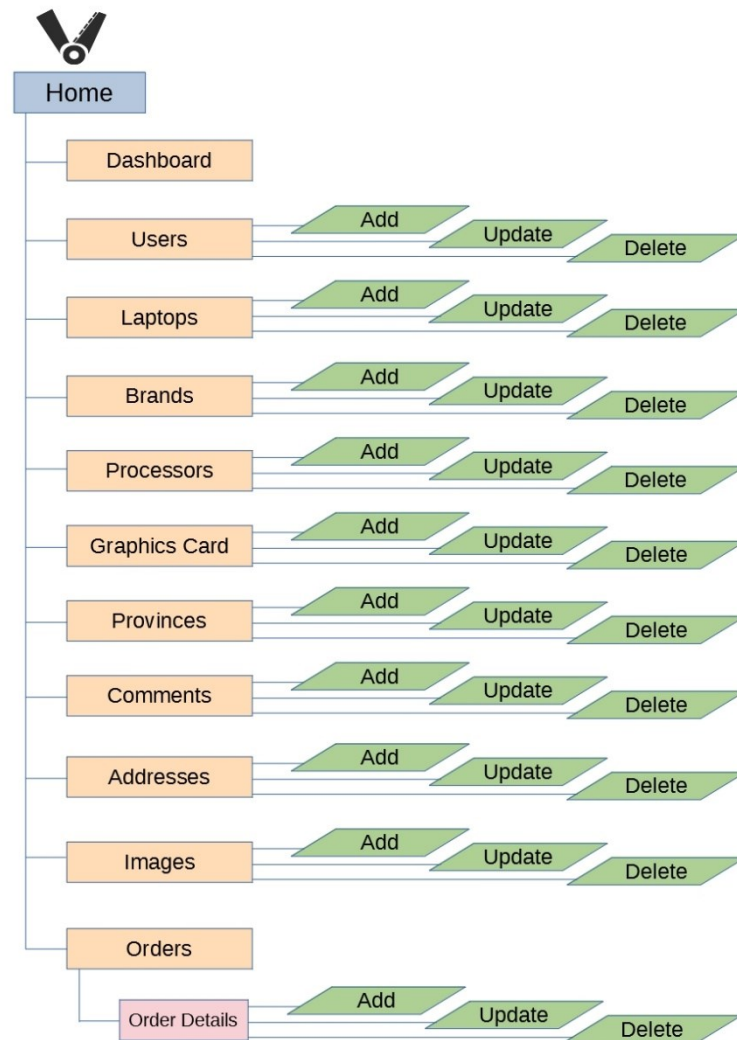
Admin: Admin User would be able to explore all of the pages of the site as well as the admin pages on which admin would be able to post new offers of laptop and also can add a new item to the site. Admin would be able to perform CRUD operations for every database table.

5. Sitemaps:

Sitemap - User:



Sitemap - Admin:



Admin Sitemap
VIDIA
Prepared By: Sandeep Pardeshi
Version: 1.0
Date: 2020-11-29

Legends

Index	Navigation	Sub Page	Links
-------	------------	----------	-------

6. Server Setup:

As part of an e-Commerce Project, we did the following setup and configured a secure LAMP stack to host our website. We were given a hosted domain on uwpace.ca and a server space on Ubuntu server (version: Ubuntu 20.04.1 LTS) with a username `team2` and FQDN as `vidia.uwpace.ca`. The following requirements were implemented in the Server setup:

- ❖ Logged in on ssh using Git Bash with `ssh -l team2 vidia.uwpace.ca`
- ❖ Updated and upgraded the server packages using `"sudo apt update"` and `"sudo apt upgrade"`.
- ❖ Installed Apache on the server using `"sudo apt install apache2"`
- ❖ Installed, started and enabled Mariadb Server using `"sudo apt install. mariadb-server"`, `"sudo systemctl start mariadb.service"`, `"sudo systemctl enable mariadb.service"`.
- ❖ Installed zip unzip, git, php and required extensions to run Laravel using following commands -
 - `sudo apt install zip unzip`
 - `sudo apt install git`
 - `sudo apt install php php-cli php-curl php-common php-gd php-mysql php-mbstring php-zip php-xml`
- ❖ Re-started apache2 using `"sudo systemctl restart apache2.service"`.
- ❖ Created bin directory on root and inside bin installed composer through scripts.
- ❖ On root did `source .profile`
- ❖ Cloned Laravel folder as Vidias from GitHub under `/var/www` and did `composer install`
- ❖ Did all steps required for Laravel folder like creating and editing `.env` file, created database, user with its password and granted all access to it, created vhost file and given the required details in it, enabled site and reloaded apache2, and enabled rewrite.

7. Security Implemented:

Implemented server security along with rights access on laravel folder and specific folders inside it and installed a firewall with rules using the following steps -

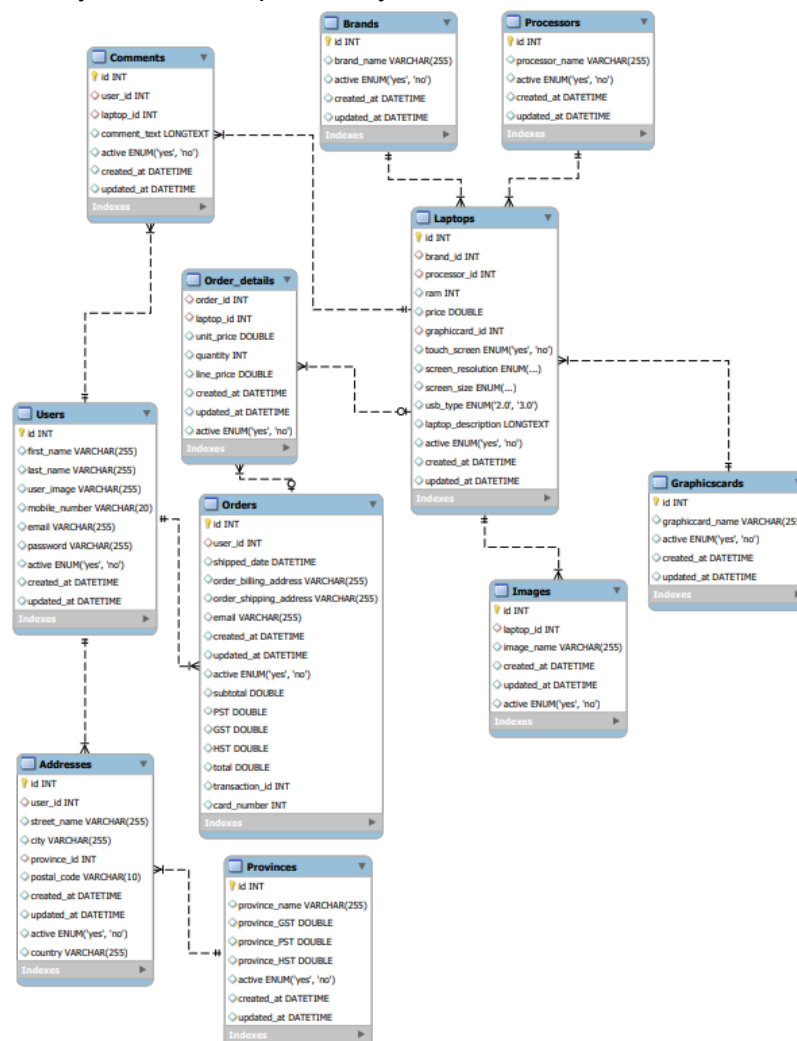
- ❖ Ownership for user and group on repository folder `Vidias` given to `team2` and file permission rights as `755` recursively.
- ❖ Inside the repository, the laravel folder (`vidias`) is owned by `team2` recursively. Inside the laravel folder, the following steps are done with ownership and file permissions setup:

- `vidias/database` owned by team2 as user and group, with 755 permissions recursively
- `vidias/storage` owned by user as team2 and group as www-data with 775 permissions recursively.
- `vidias/bootstrap/cache` owned by user as team2 and group as www-data with 775 permissions recursively.
- `vidias/tmp` folder created and owned by user as team2 and group as www-data with 775 permissions recursively.
- Properly secured PHP as -
 - For public folder:
 - `AllowOverride All`
 - `Options -Indexes +FollowSymLinks`
 - For upload folder:
 - Established link between public and storage folder and created `.htaccess` under `/var/www/Vidias/vidias/storage/app/public/images` where the `php_flag engine 0` is written to stop any .php file to execute.
 - `.htaccess` is owned by team2 as user and group, and the file permission is given as 644 on `.htaccess`
 - The following `php_admin_value`s set outside any Directory declarations:
 - `php_admin_value log_errors 1`
 - `php_admin_value error_log "/var/www/Vidias/vidias/php_error.log"`
 - `php_admin_value open_basedir "/var/www/Vidias/vidias"`
 - `php_admin_value temp_uploads_directory "/var/www/Vidias/vidias/tmp"`
 - `php_admin_value file_uploads 1`
- ❖ An Uncomplicated Firewall or UFW is installed on the server using `"sudo apt install ufw"` and the following commands were executed. Link followed <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-18-04>
 - Setting up default policies with `"sudo ufw default deny incoming"` and `"sudo ufw default allow outgoing"`
 - Allowing SSH connections using `"sudo ufw allow ssh"`, `"sudo ufw allow 22"`
 - Enabling UFW with `"sudo ufw enable"`
 - Allowed only two ports after enabling ufw using `"sudo ufw allow 80"` and `"sudo ufw allow 443"`
- ❖ MySQL/MariaDB configured for a secure production environment using `mysql_secure_installation` on the shell prompt where the root password was also created in the process. Link followed - https://mariadb.com/kb/en/mysql_secure_installation/

- ❖ SSH root access disabled by editing `/etc/ssh/sshd_config` file for `PermitRootLogin no` and running `service ssh restart`

8. Database:

The database of the Vidia e-Commerce website showcases the relationships between laptops, users, and orders tables. Laptops table is the primary table, which has three foreign keys. All tables comply with 3NF normalization. The relationship between tables Brands, Processors, and Graphicscards with the Laptops table are all one-to-many (1:M). In comparison, the relationship between the Laptops table with Comments and Images tables is one-to-many (1:M). The relationship between Orders and Laptops is many-to-many (M:N), so there is an intersect table called Order_details to normalize this relation to comply with 3NF. There is a Provinces table containing data columns called province_name, province_PST, province_GST, and province_HST. When the address is outside of Canada, the users will not be charged GST and PST. As a result, the province_PST and province_GST will be zero in this case. There is historical data in the Orders table showing the full information of that order once it is placed. And this historical data will not be determined by a relationship to timely data.

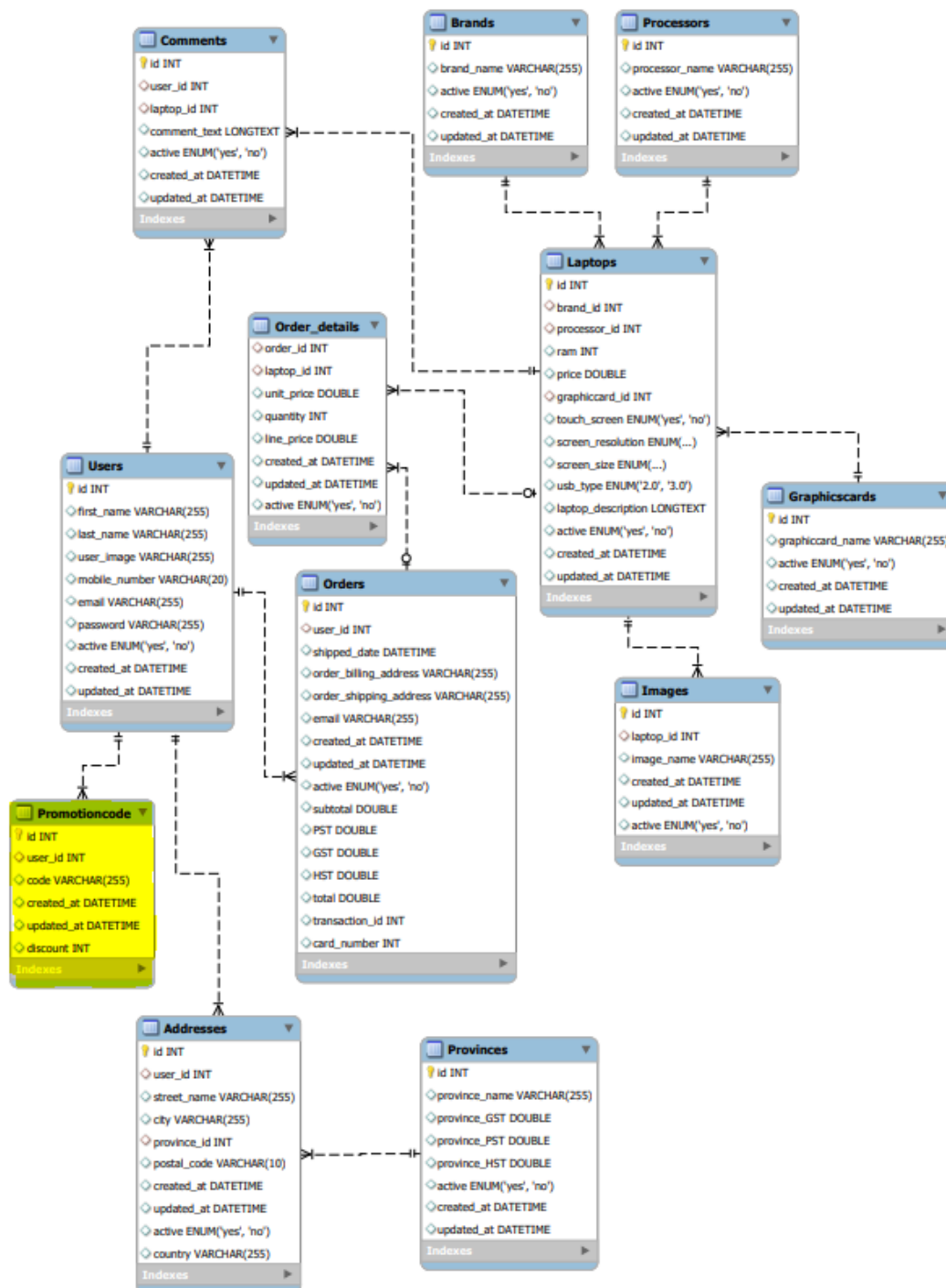


9. Value Adds:

A promotion code can be entered during the checkout process to apply a discount to the order before taxes. A Promotioncode table is added into the database to store the discount percentage and the string for the promotion code and the user_id which this promotion code is applied to.

10. Revised ERD with Value Adds:

Promotioncode table has a foreign key from Users table. The relationship between Users table and Promotioncode table is one-to-many (1:M).



11. Our Team:

We have a team of experienced developers who are skilled in their roles and have worked for multinational companies in the past. We hold experience in developing e-commerce websites, and our team has worked with big clients. Our team is passionate about the work, and we work to deliver the best in class.

Our team:

Names	Position	Experience
Huagng Junyan	Database Administrator, and Programmer	1 Year
Kulshrestha Anu	Server Administrator, and Programmer	1 Year
Lwabanji Mika Alain	HTML, CSS, Content writer, and Programmer	1 Year
Nnorom Chinedu	HTML ,CSS and Programmer	1 Year
Pardeshi Sandeep	Project manager, and Programmer	1 Year
Parmar Sagar	Lead Programmer, and Git Manager	1 Year
Xu Ziyang	Graphic design, and Programmer	1 Year