**1. Introduction:**

It is a website named "CODE PORT". The main purpose of this portal is for User to Ask and answer question. We have developed this website in PHP as Front End with the MYSQL as a back end and various CSS as well as to fulfil our website's “CODE PORT”. This project provides Share knowledge and solve coding challenges. When you log in it, we can access the process of different question. This website saves your times and power. It is very easy to handle and it is very simple. The project entitled “Code Port” enables a user to provide a facility of solving their doubts through online.

**1.1 Existing System:**

Creating a Cort Port in Laravel, a web development framework, involves setting up your project, designing your database, and writing code to make everything work. Laravel makes it easy to manage users, like signing up and logging in, thanks to its built-in tools. You'll create pages where users can ask questions, answer them, and vote on the best answers. Laravel helps organize your code so that everything stays neat and easy to understand. It also helps you search for questions and answers quickly. When users type in keywords, Laravel finds the relevant stuff fast. For how it looks, you'll use Blade, which is like a fancy way to write HTML. Blade makes it simple to create web pages that look good and work well. Of course, you'll also need to make sure your website is safe from hackers. Laravel has built-in tools to help with that too, so you can feel confident your users' information is secure.

**1.2 Need for the New System:**

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was tedious and lengthy. The records were never used to be in a systematic order. There used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. It was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update records. To overcome such reasons, a computerized system is created.

**1.3 Objective of the New System:**

**User Engagement:** Encourage active participation by allowing users to ask questions, share knowledge through answers, and vote on the most helpful contributions.

**User Management:** Provide a seamless experience for users to register, log in, and manage their profiles, ensuring smooth interaction with the platform.

**Content Organization:** Implement features for tagging questions, searching for relevant content, and filtering information based on user preferences, facilitating efficient information retrieval.

**Security:** Ensure the safety of user data and the integrity of the platform by implementing robust security measures, including user authentication, input validation, and protection against common web vulnerabilities.

**Scalability and Performance:** Design the application to handle a growing user base and maintain optimal performance, utilizing Laravel's caching mechanisms and optimization techniques.

**Community Building**: Foster a sense of community among users by providing tools for communication, collaboration, and moderation, facilitating productive interactions and knowledge sharing.

**1.4 Problem Definition:**

In the current websites might have information spread out or not very engaging features. So, we want to create a platform where it's easy to find answers and where users can interact with each other in a friendly way. With our Stack Overflow clone, we aim to bring all the helpful info together, make searching for it simple, and encourage users to help each other out. Plus, we'll make sure it's safe to use and can handle lots of people using it at once. Our goal is to make a website where anyone can get the help they need and feel like part of a supportive community.

**1.5 Core Components:**

**User Management:** This component handles user authentication, registration, login, and profile management. It ensures that users can securely access the platform and maintain their personal information.

**Question and Answer System:** This component allows users to ask questions, provide answers, and interact with each other's content. It includes features such as posting questions, commenting on answers, and voting on the best solutions.

**Tagging and Categorization:** Tags are used to categorize questions and make them easier to find. This component enables users to add tags to their questions and filter content based on specific topics or keywords.

**Search Functionality:** The search component enables users to find relevant questions and answers quickly. It allows users to search by keywords, tags, or categories and delivers accurate results in real-time.

**Voting and Reputation System**: Users can vote on questions and answers to indicate their quality and helpfulness. A reputation system tracks users' contributions and assigns reputation points based on their activity and the votes they receive.

**Notification System:** This component notifies users about new answers, comments, votes, or other relevant activities on their questions or answers. It ensures that users stay informed and engaged with the community.

**Moderation Tools:** Moderators have access to tools for managing content, handling flags, and enforcing community guidelines. These tools help maintain the quality and integrity of the platform by addressing spam, inappropriate content, or violations of community rules.

**1.6 Project Profile:**

|  |  |
| --- | --- |
| Project title | Code Port Website |
| Group no: | 30 |
| Duration | 3 months |
| Frontend | PHP, HTML, CSS |
| Backend | LARAVEL |
| Database Server | MySQL |
| Software requirement | VS Code |
| Operating System | Windows 11 |
| Developed by | IET-10 Kshama Gandhi  IMS-26 Purvi Makwana |
| Internal Guide | Professor Jayshree Dasa |
| Institute Name | LJ Institute of computer applications |

**1.7 Assumptions and Constraints:**

* The code should be free with compilation errors/syntax errors.
* The product must have an interface which is simple enough to understand.

**1.8 Advantages and Limitations of the Proposed System:**

* **Advantages:**
* The Code Port will be able to post his ads on the website.
* The customer will be able to access that post and contact the renter.
* The renter and customer documentation/dealing/payment will be done face to face to avoid spam.
* This system is fully functional and flexible.
* It saves a lot of time, money and labour.
* First the customer has to do registration.
* Second if the customer had already registered himself/herself then he/she can continue booking in his/her own account by giving his/her customer id or mail id.
* Thirdly, the customer can amend details or update details.
* Maintenance is easy and performance is good.
* It reduces the time complexity.
* **Limitations:**
* Inconvenience for customer
* Manual record-keeping
* Lack of transparency
* Security risks
* Time-Consuming
* High administrative burden

**2. Requirement Determination and Analysis:**

**2.1 Requirement Determination:**

➢ **Hardware**

The starting point in an equipment decision process is the size and capacity requirements. One particular computer system may be appropriate for one workload and inappropriate for another.

Systems capacity is frequently the determining factor.

Features considered were:

* Internal memory size
* Speed of the system processing
* Characteristics of display
* Types and numbers of auxiliary storages units that can be attached.
* System support and utility software provided.
* Software needs the minimum configuration for specific use.

➢ **Software**

For determination of system software to be used for preparation of the system, the following considerations have been investigated:

* The transactions and data to be handled
* The reports, documents and other outputs to be produced.
* The files and database to be prepared
* The transaction and master file needed to maintain the system, The volume of transactions to be processed
* The flexibility of a software system should include the ability to meet changing requirements and varying user needs.
* Taking into view the above, the following hardware and software are suggested for the preparation and implementation of the project.

**➢ Hardware Requirements**

* Pentium IV 2.0 GHz or higher processor.
* 1 GB or higher capacity RAM.
* 60 GB or higher capacity Hard Disk
* Color monitor

➢ **Software Requirements**

* Operating system: Windows 10
* Microsoft Visual Studio code / notepad
* Software: Xampp (8.1.6)

**➢Functional requirement:**

1. Admin:

* The admin manages the whole data. Admin can insert new details, modify flight changes and remove a current detail Admin has their user name and password they can also Change his/her profile by using password.

1. User service providing:

* They are the users of the system. They register themselves with the system.
* They can register, give feedback and contact if they have any queries.

**➢Non-functional requirement:**

1. Reliable:

➢ A software is much ecofriendly, many places the direction and the process Has been added put up to help a users for travelling in easier and faster way.

1. Security:

* Secure to a greater extended though the admin and User enter their id and Password or registration through the some from, still the user can never enter the admin’s account.
* Certain important information is only visible and assessed by the admin only.

3. Interface:

➢ The software will be compatible every windows platform. Operating system like windows XP, vista, windows7, windows8, windows10, windows11 will support it.

1. Platform: Visual Studio code / notepad
2. Front end: PHP
3. Back end: MYSQL

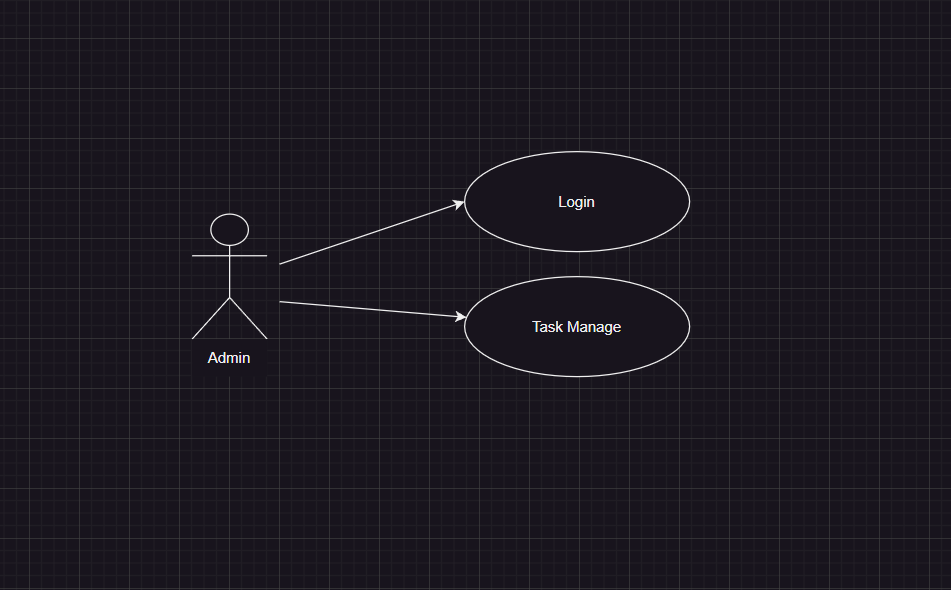
**2.2 Targeted Users:**

**Admin:** Admin module include user management, content moderation, analytics, settings configuration, security features, and support tools for maintaining a safe and engaging platform environment.

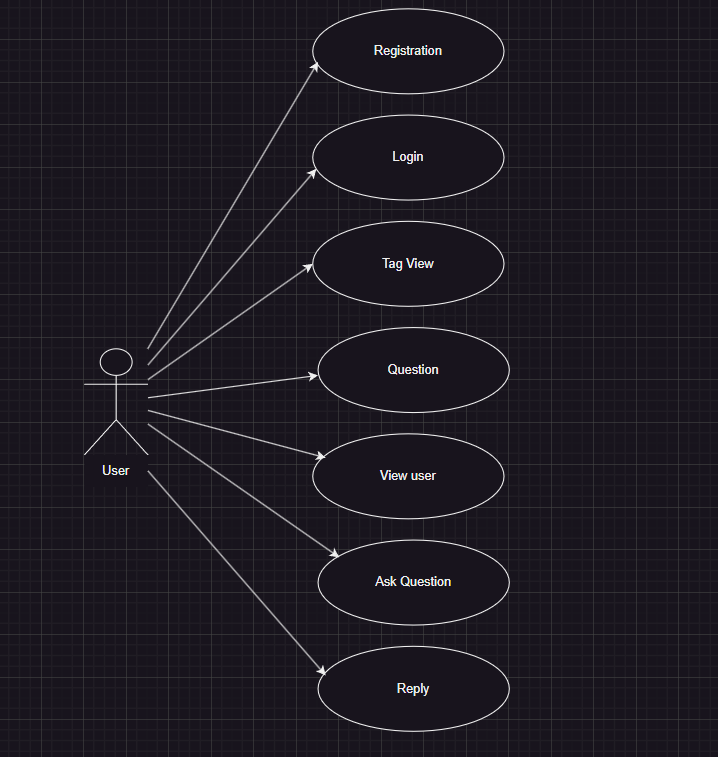
**User:** User module include asking and answering questions, voting on content, engaging in discussions, and managing their profiles and notifications for a seamless and interactive experience within the community.

**3. System Design**

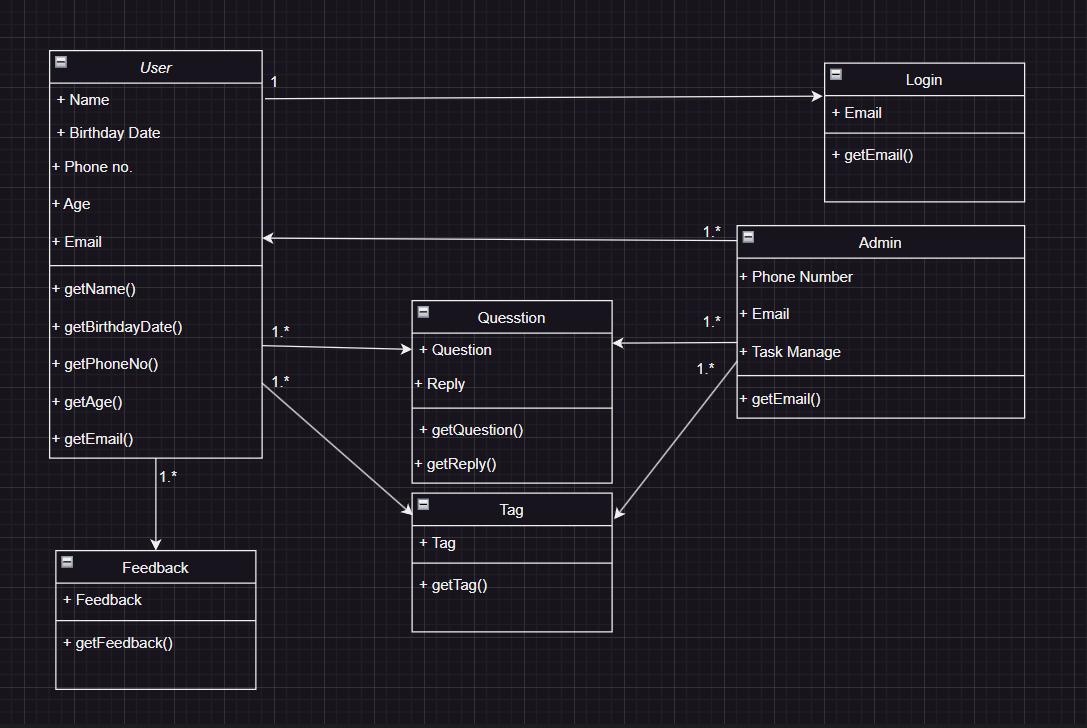
* 1. **Use Case Diagram**
* **Admin:**



* **User:**



* 1. **Class Diagram**

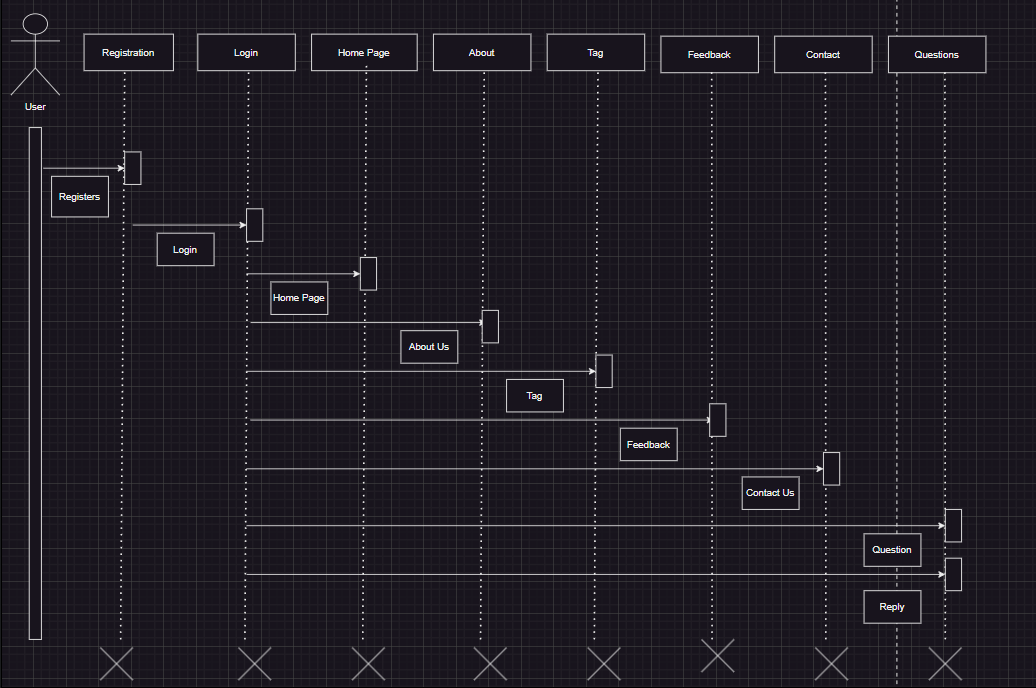


**3.3 Interaction Diagram**

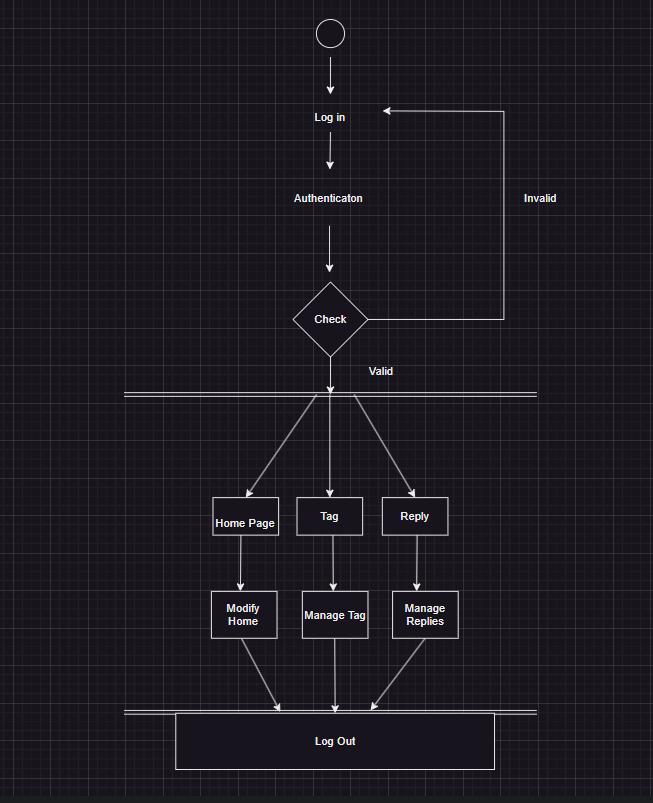
* **Admin:**



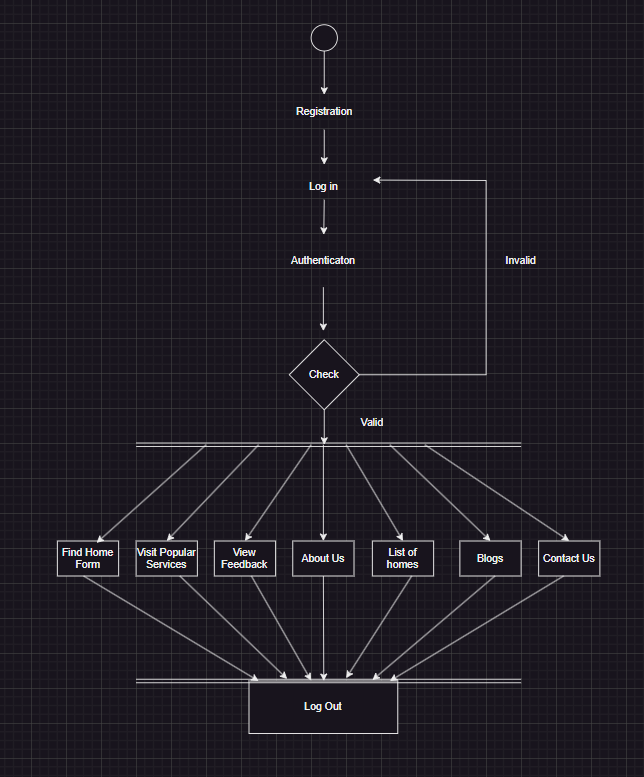
* **User:**



* 1. **Activity Diagram**
* **Admin:**



* **User:**



* 1. **Data Dictionary:**
* **Table: Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| A\_ID | Int | PrimaryKey | Enter user id |
| A\_name | Varchar(50) | Not Null | Enter User name |
| Email | Varchar(50) | Not Null | Enter email id |
| Password | Varchar(50) | Not Null | Enter Password |

* This table is list of Admin Informations.
* **Table : Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| U\_id | Int | PrimaryKey | Enter My JobID |
| U\_name | Varchar(50) | Not Null | Enter user name |
| U\_email | Varchar(50) | Not Null | Enter Email id |
| B\_date | Datetime | Not Null | Enter Birth date |
| U\_age | Int | Not Null | Enter Age |
| P\_no | Int | Not Null | Enter phone number |
| Password | Varchar | Not Null | Enter password |

* This table store user account.
* **Table: Tags**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| T\_id | Int | PrimaryKey | Enter Tag ID |
| T\_title | Varchar(50) | Not Null | Enter title of language |
| T\_des | Varchar(255) | Not Null | Enter tag description |
| T\_icon | - | Not Null | Insert logo of language |

* This table store Tag details.
* **Table: Questions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| Q\_id | Int | PrimaryKey | Enter Question ID |
| User\_id | Int | Not Null | Enter User ID |
| Q\_title | Varchar(50) | Not Null | Enter Question title |
| Q\_body | Varchar(255) | Not Null | Enter full Questions |
| T\_id | Varchar(50) | Not Null | Select tag |

* This table store Questions details.

* **Table: Replies**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| R\_id | Int | Not Null | Enter Reply id |
| R\_date | Date | Not Null | Enter Reply date |
| Q\_id | Int | Not Null | Select Question id |
| Reply | Varchar(255) | Not Null | Enter reply |
| U\_id | Int | Not Null | Select user id |

* This table store Replies details.
* **Table : Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Descrpition** |
| F\_id | Int | Not Null | Enter feedback id |
| F\_name | Varchar(50) | Not Null | Enter feedback user name |
| F\_email | Varchar(50) | Not Null | Enter email id |
| Message | Varchar(255) | Not Null | Enter message |
| F\_date | Date | Not Null | Enter date |

* This table store feedback detail.

1. **Development:**

**4.1. Coding Standards**

**Coding Style**

The following should be kept in mind to maintain a coding style. Such are, Layout, Format, Indentation, organizing code meaningfully, code must be readable and consistent which makes it easy to understand and maintain further.

**Formatting**

Always use comments for description.

Always use curly braces {and} in conditional statements.

**Example:**

if (condition)

{ // comment }

Order declarations within type groups based upon access modifier and their accessibility.

Public

Protected

Internal

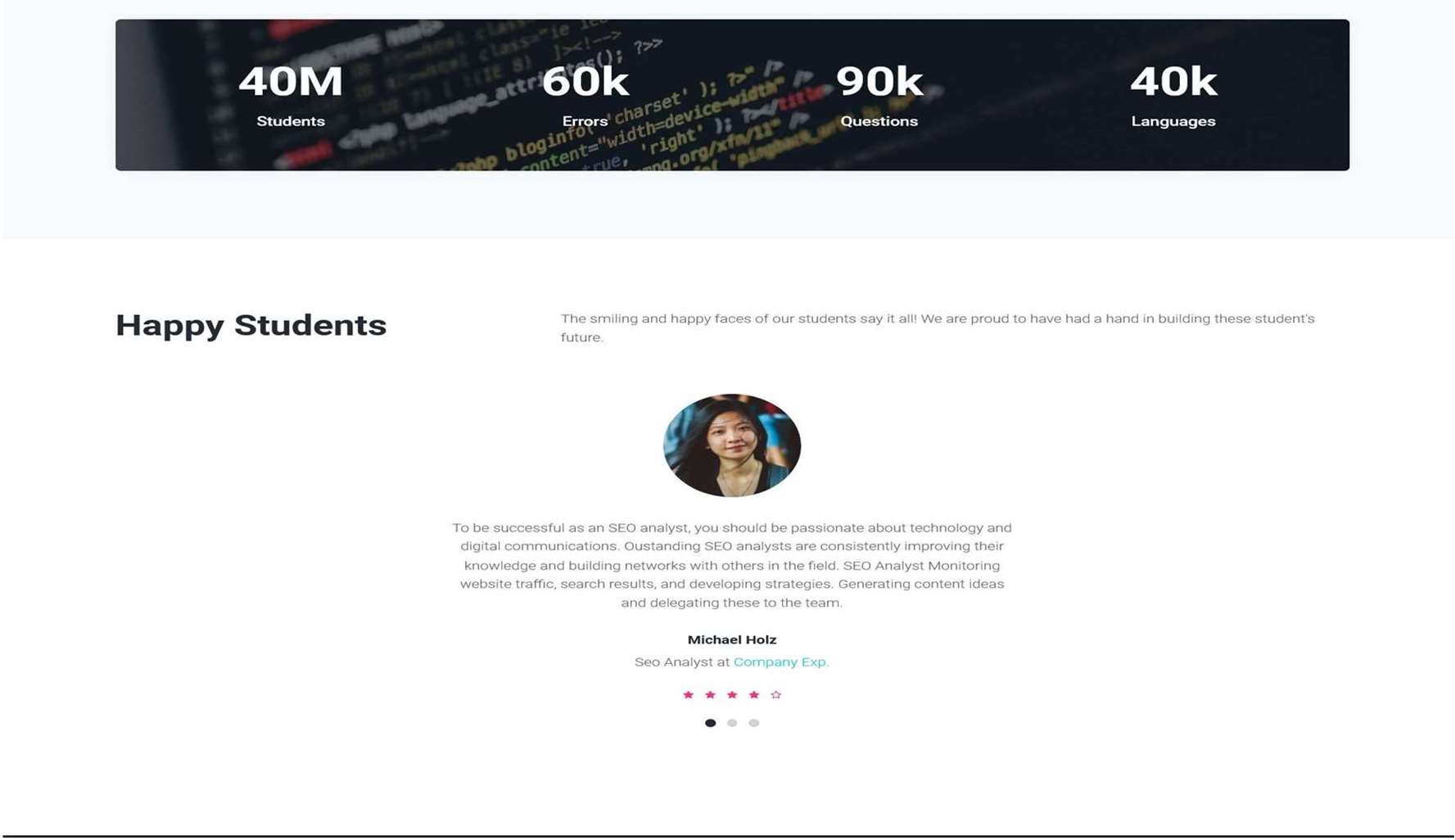
Private

Commenting

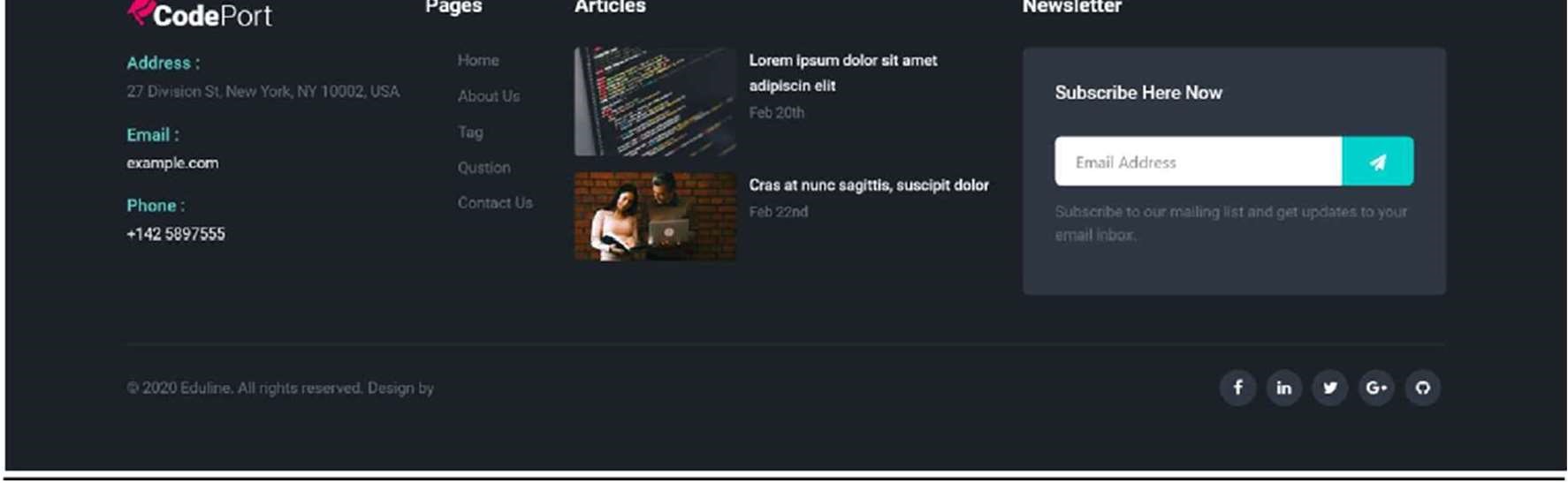
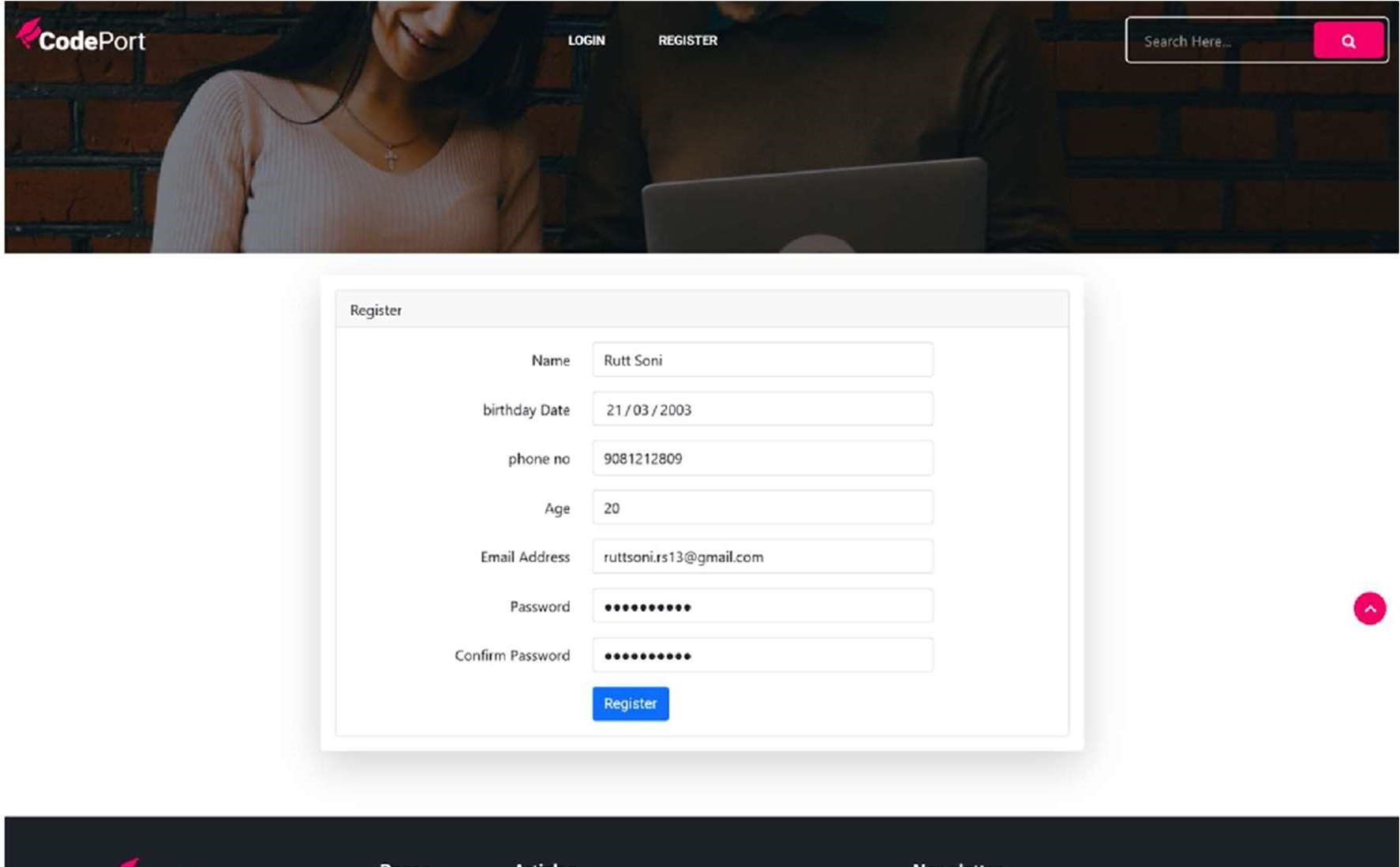
All comments should be written in U.S. English.

Use # or “””…””” but do NOT use /\* \*

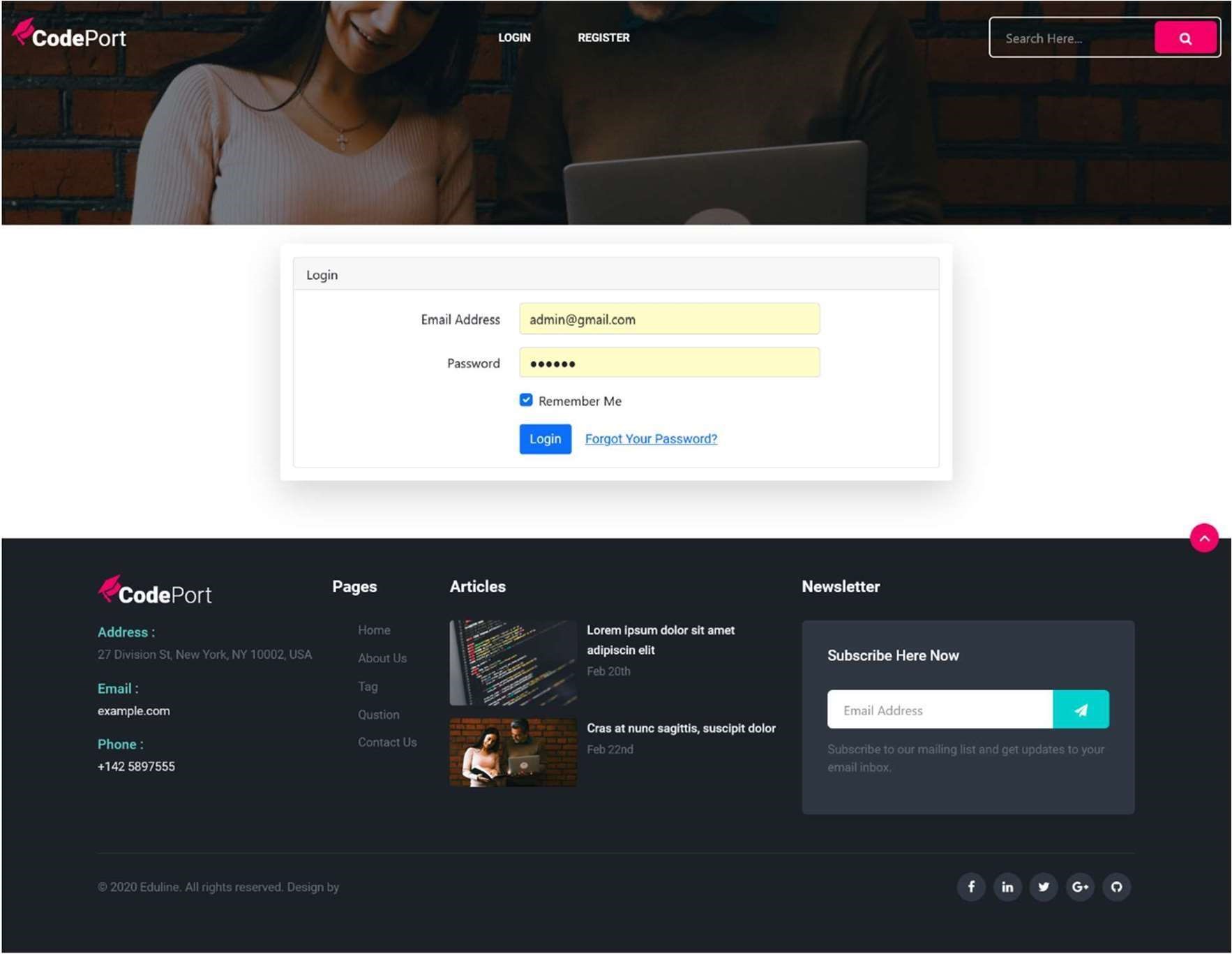
* **Screenshots:**
* **Home Page**:



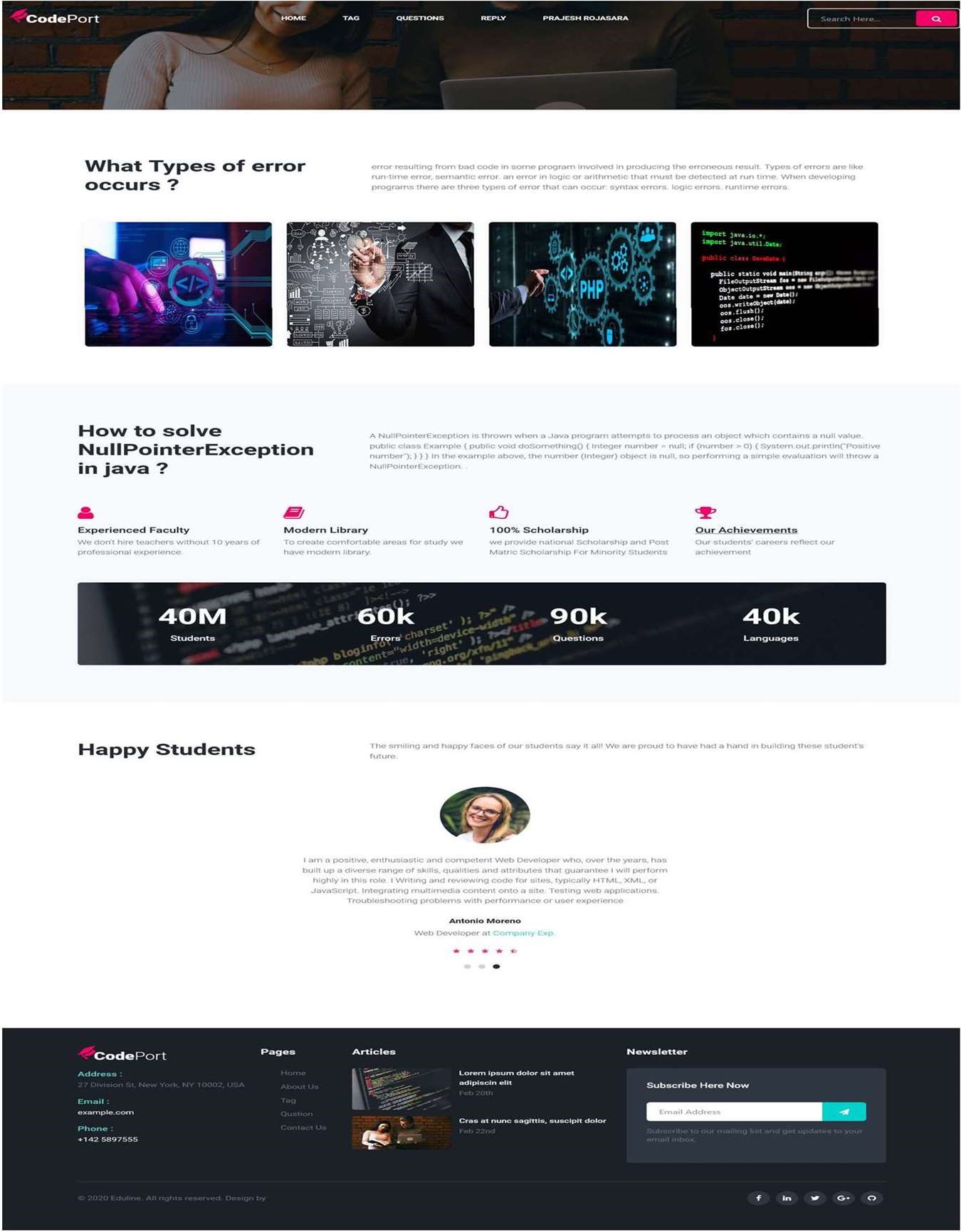
* This is our Home page.
* **Register:**



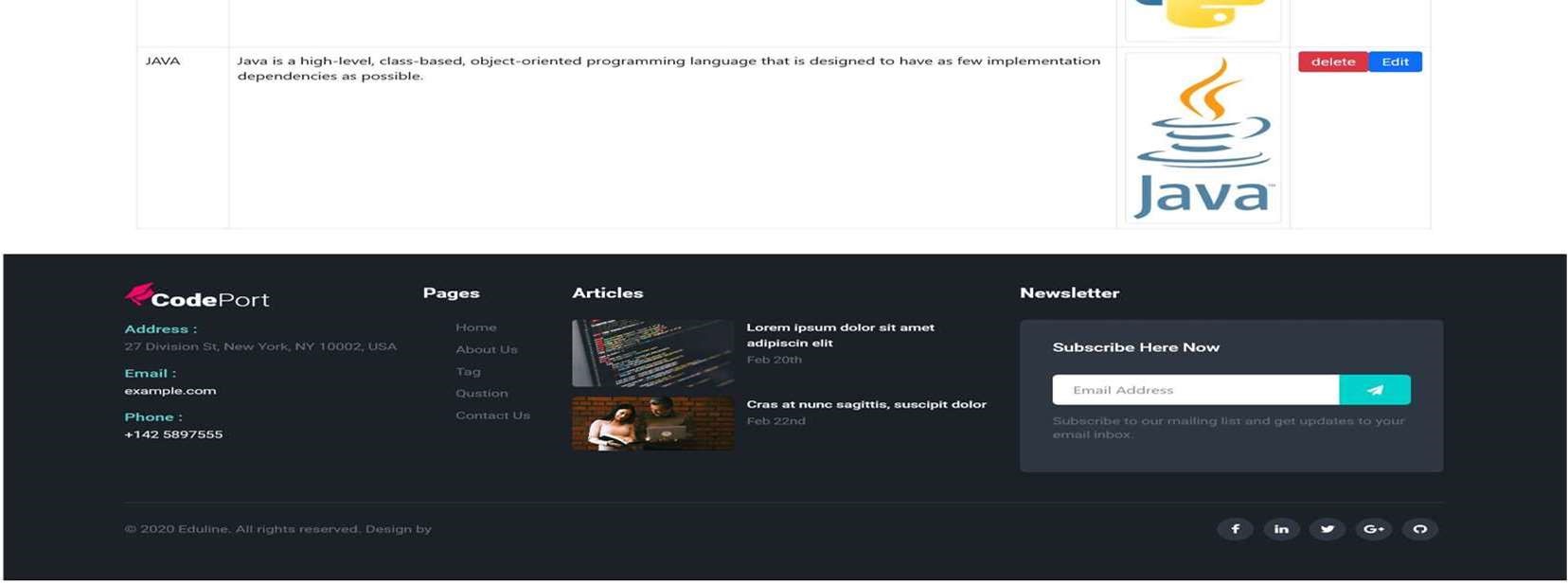
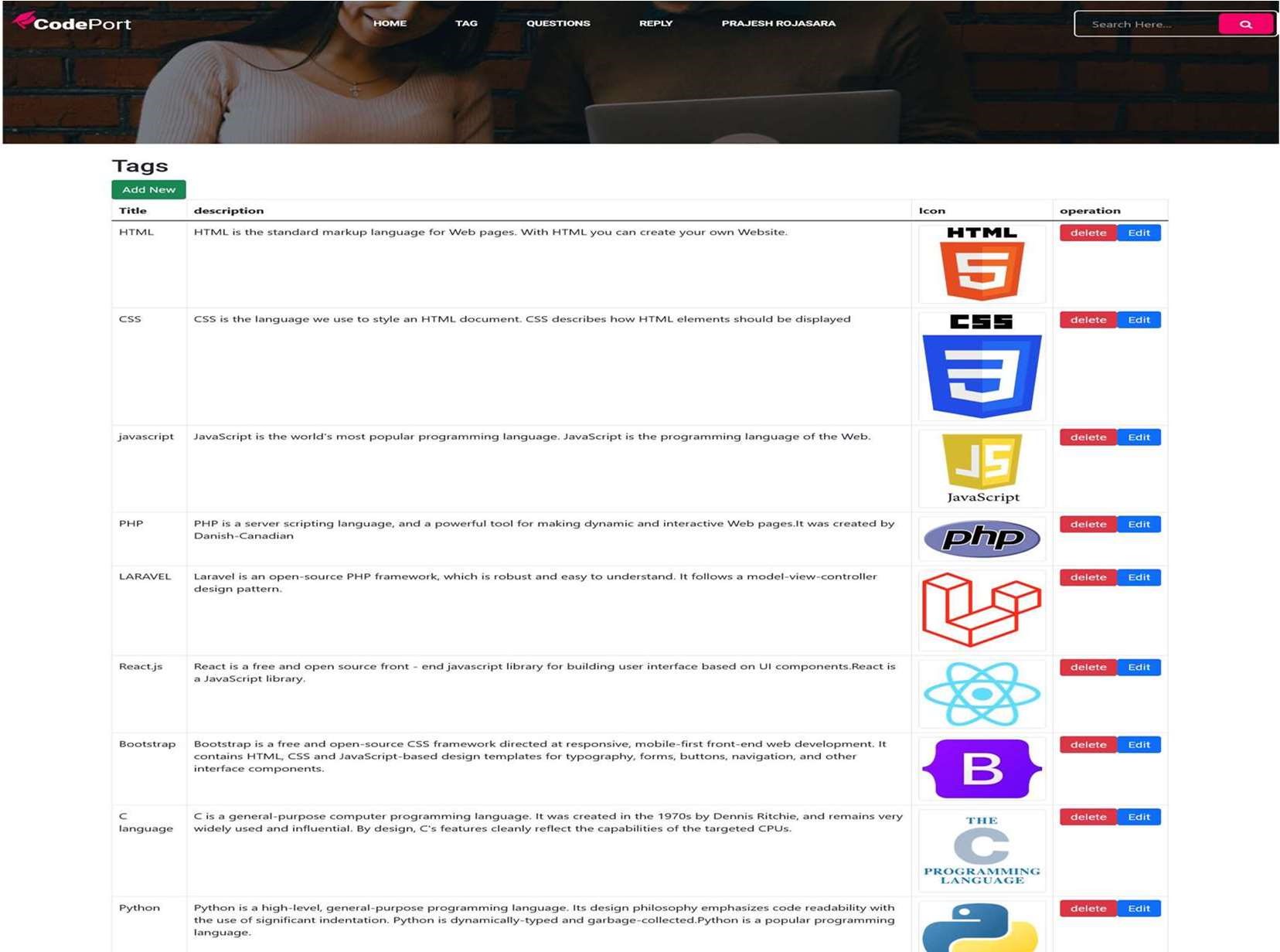
* Using Register user can be member of site As per authorization user can access only as the rights given to them.
* After being member, no one can access anything in site.
* **Admin Login:**



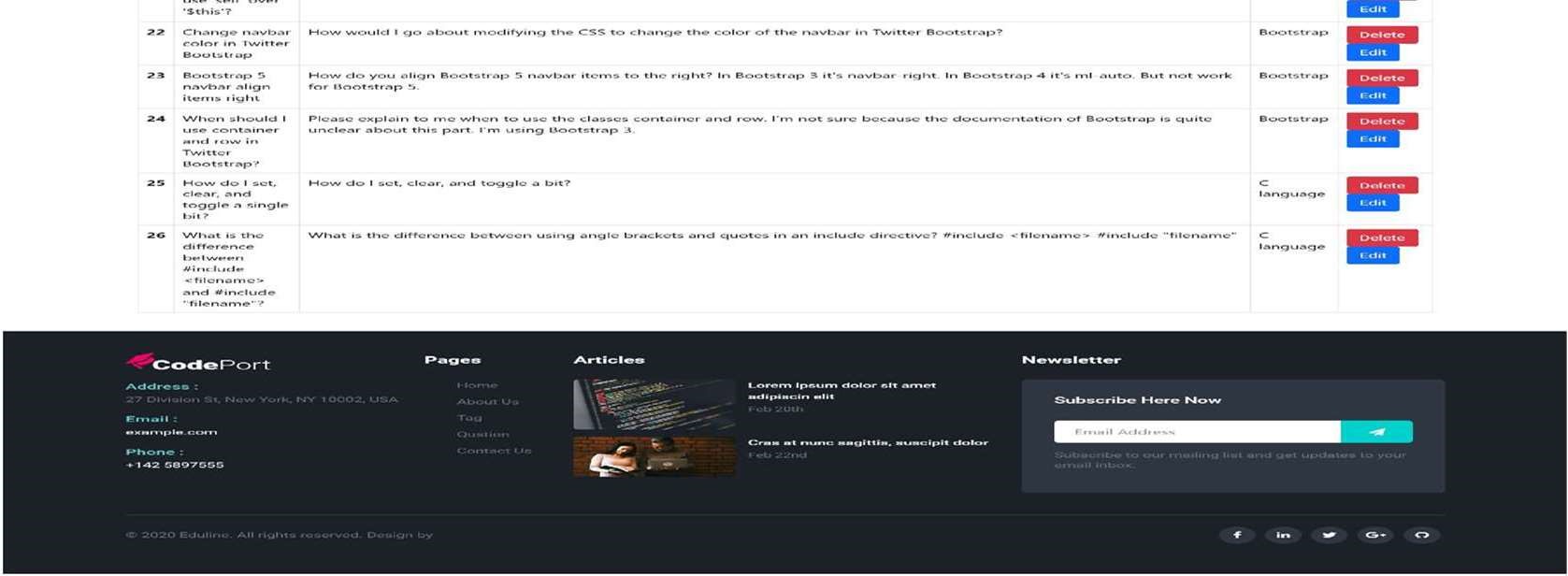
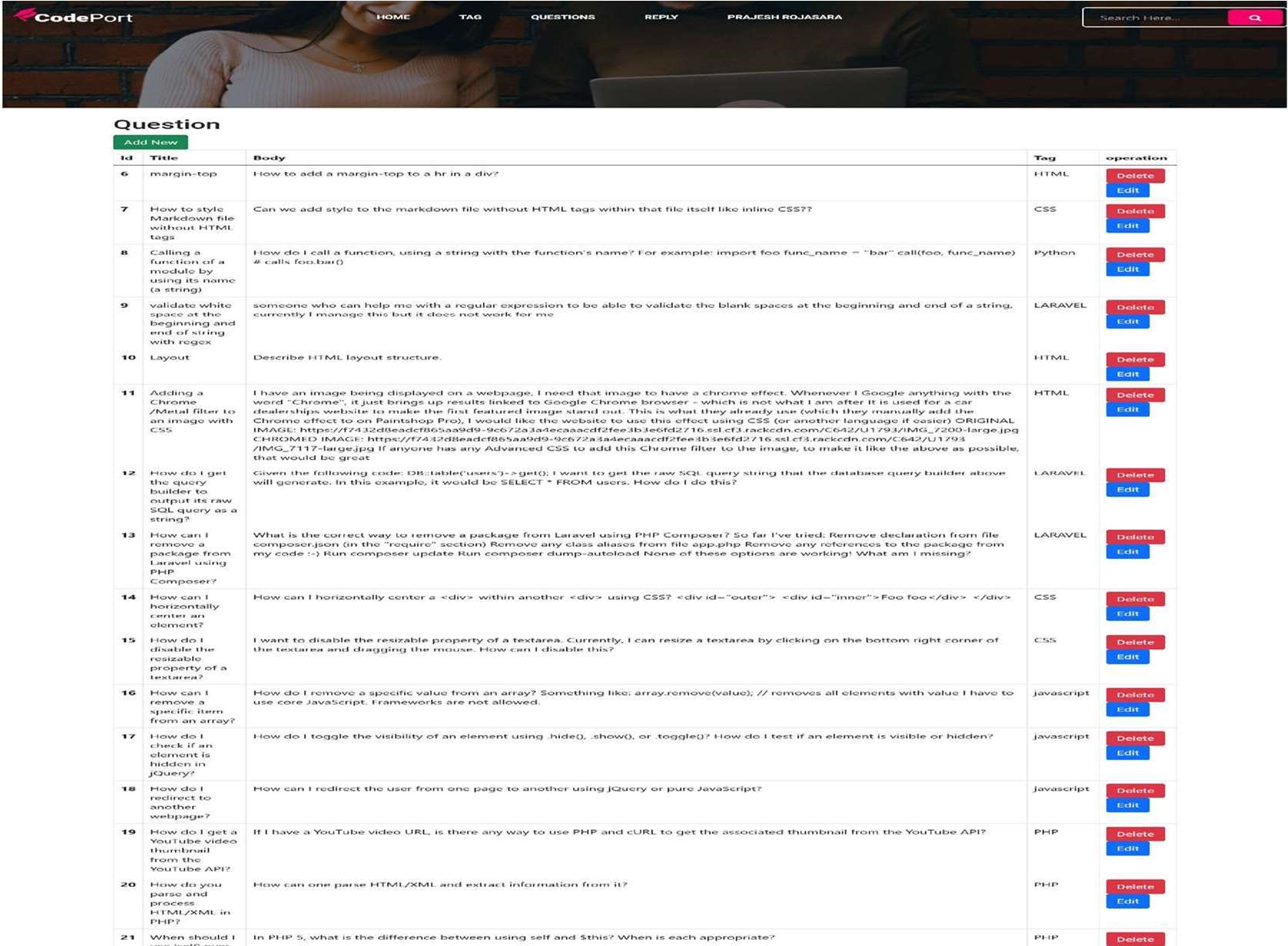
* This is login page of Admin in this page Admin can login.
* If username and password are invalid, He\she can see message “username and password are invalid”.
* **Admin Home Page**:



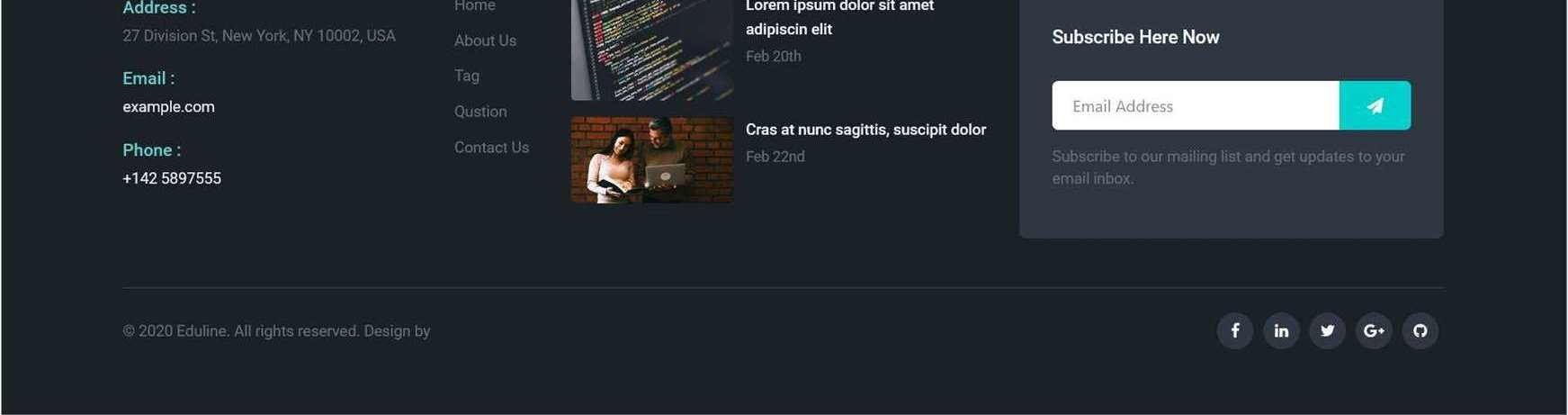
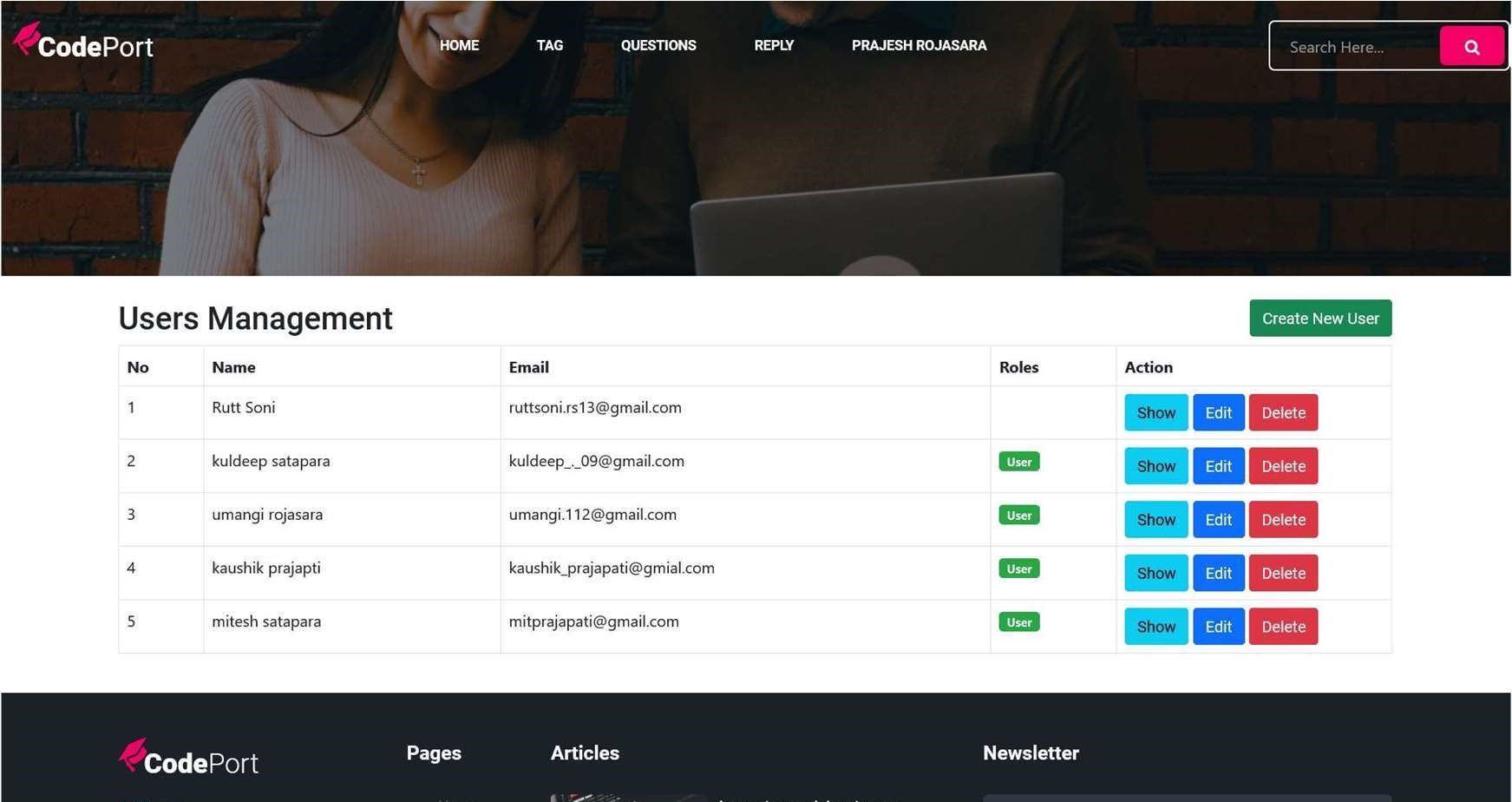
* This is Admin Home page.
* **Admin Tag Page**:



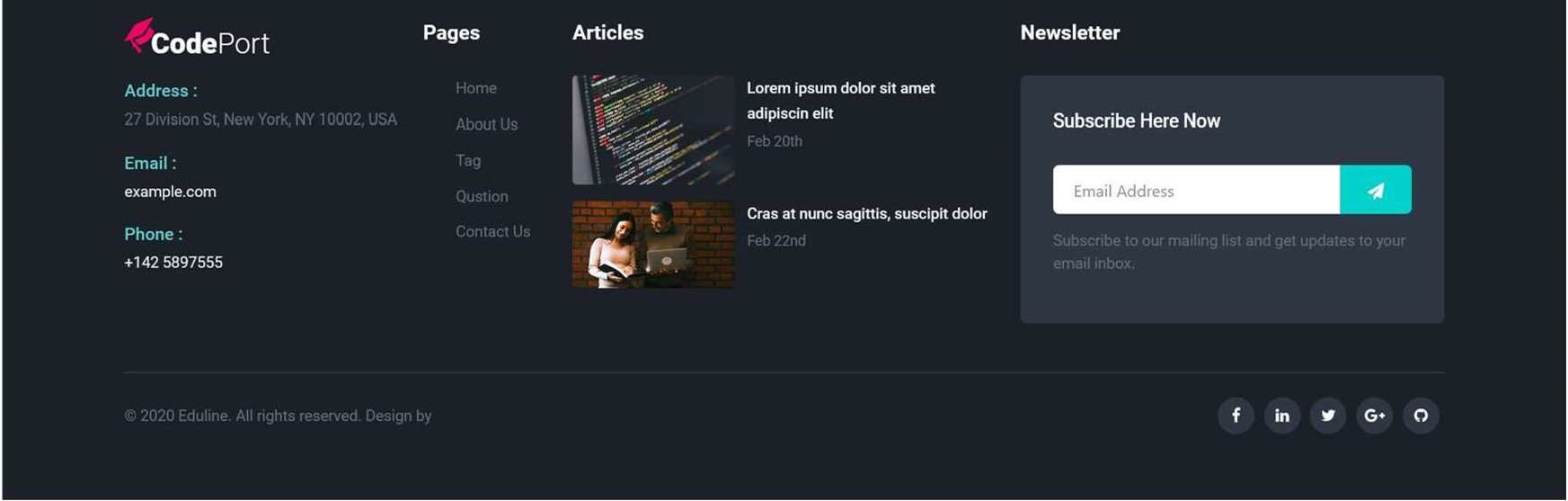
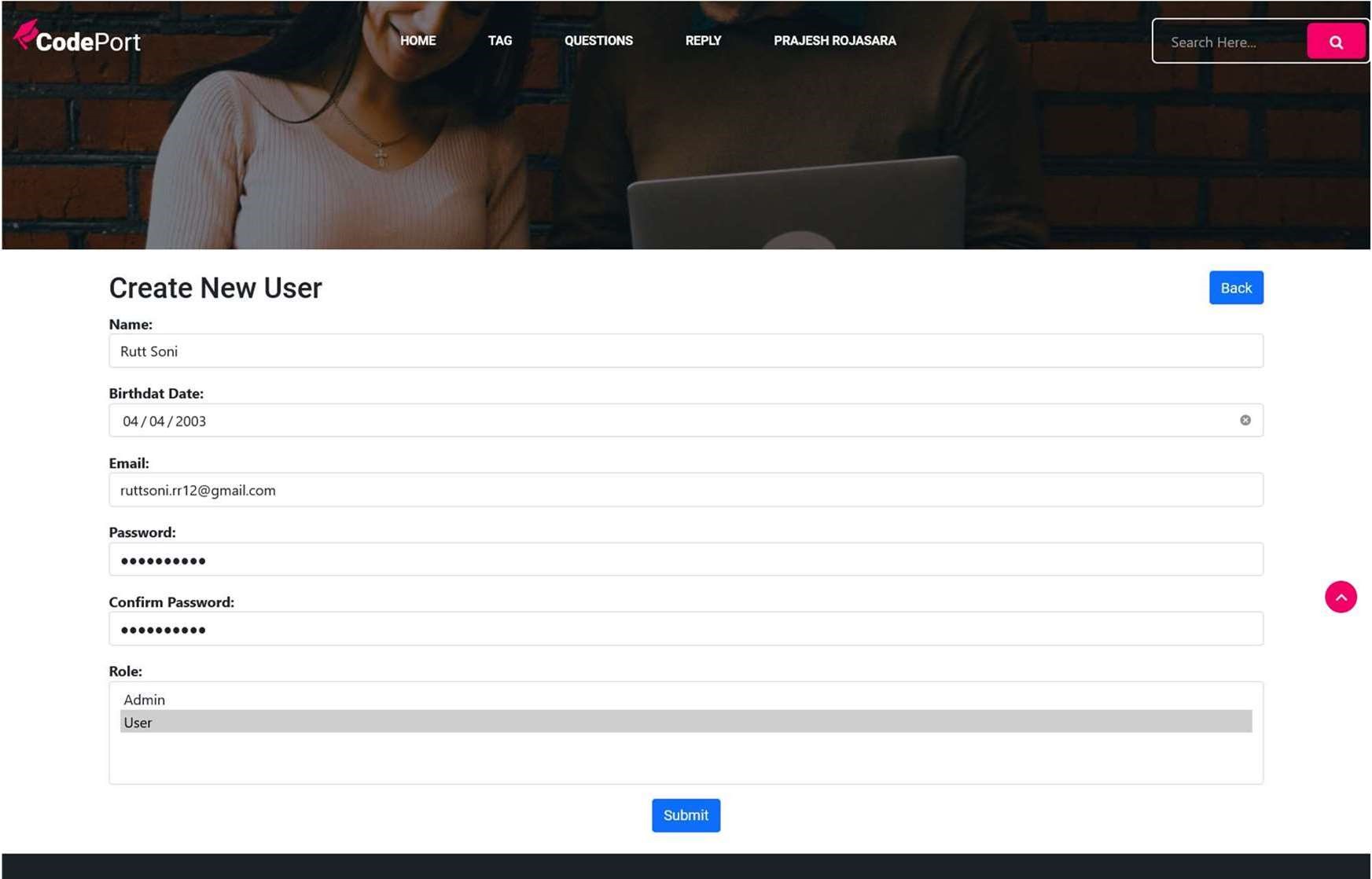
* This page is Admin Tag page where admin can add new tag, delete tag and also edit.
* **Admin Question Page**:



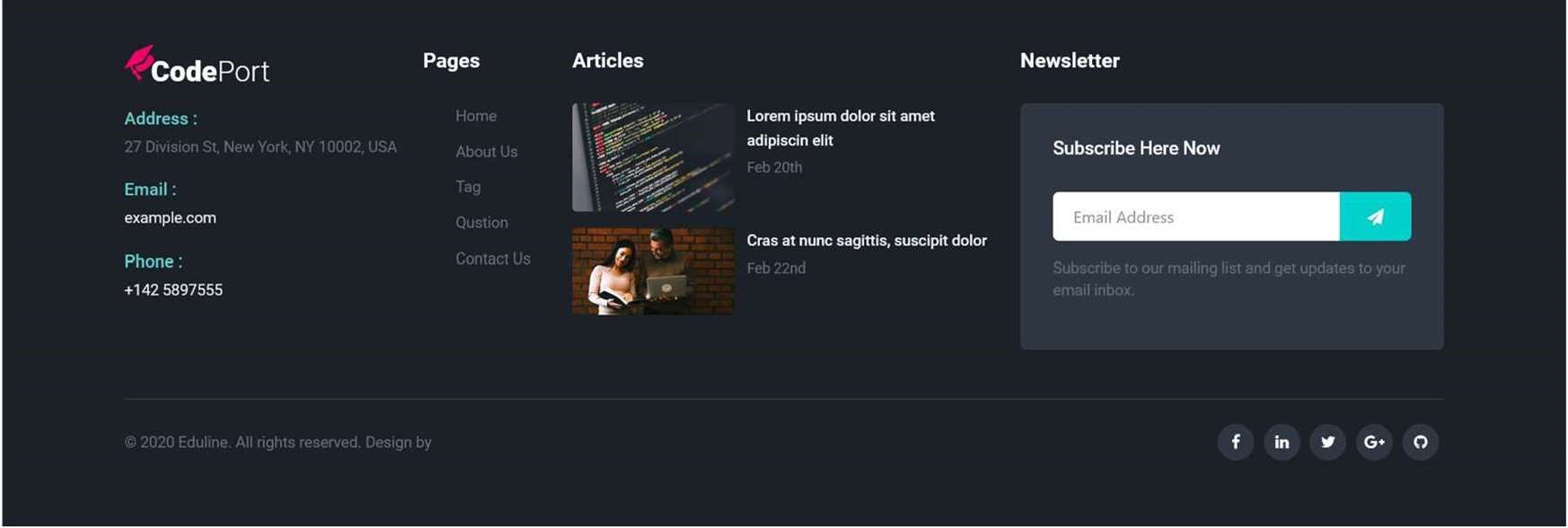
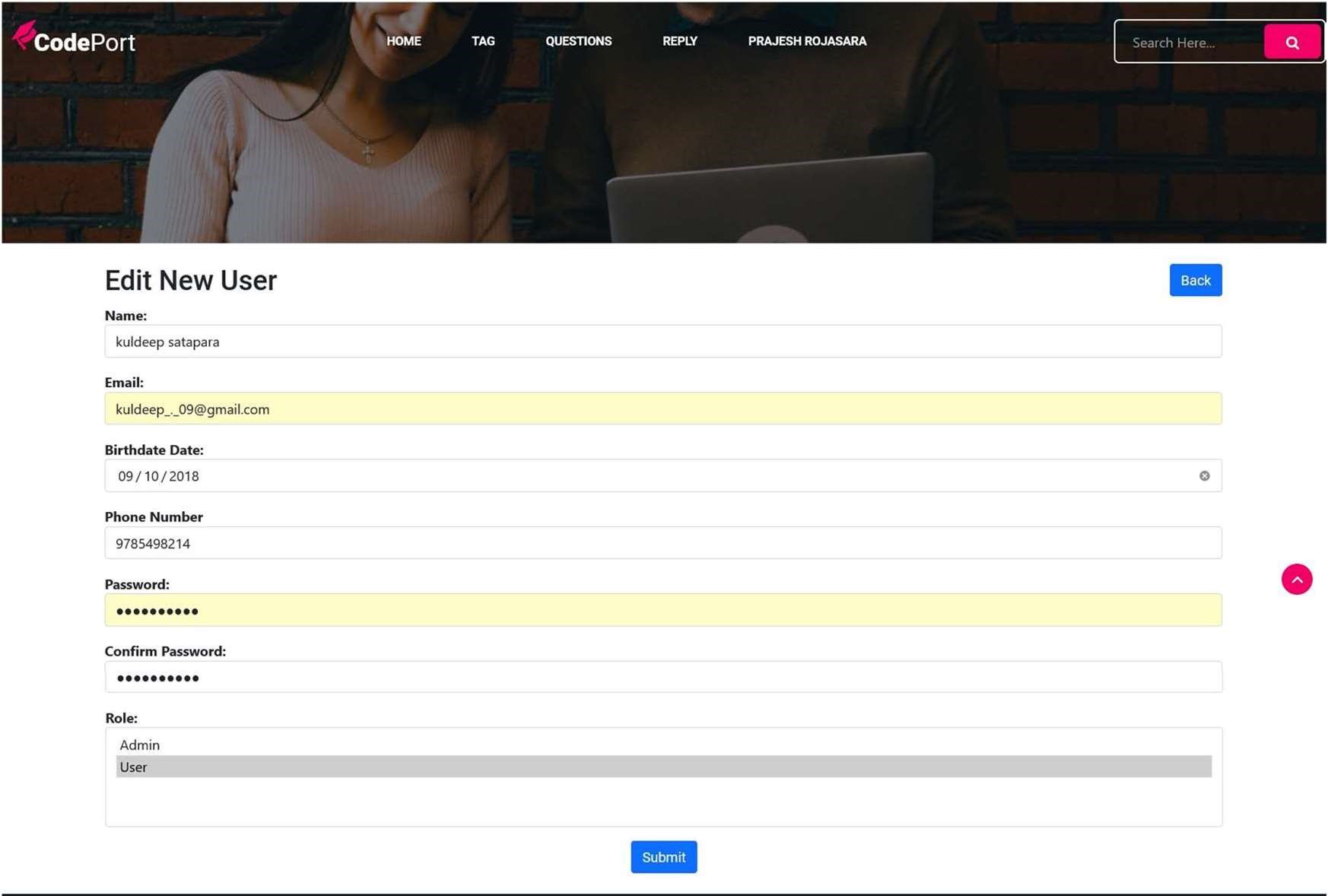
* This is Question page where User can ask their questions freely.
* **Admin User List Page**:



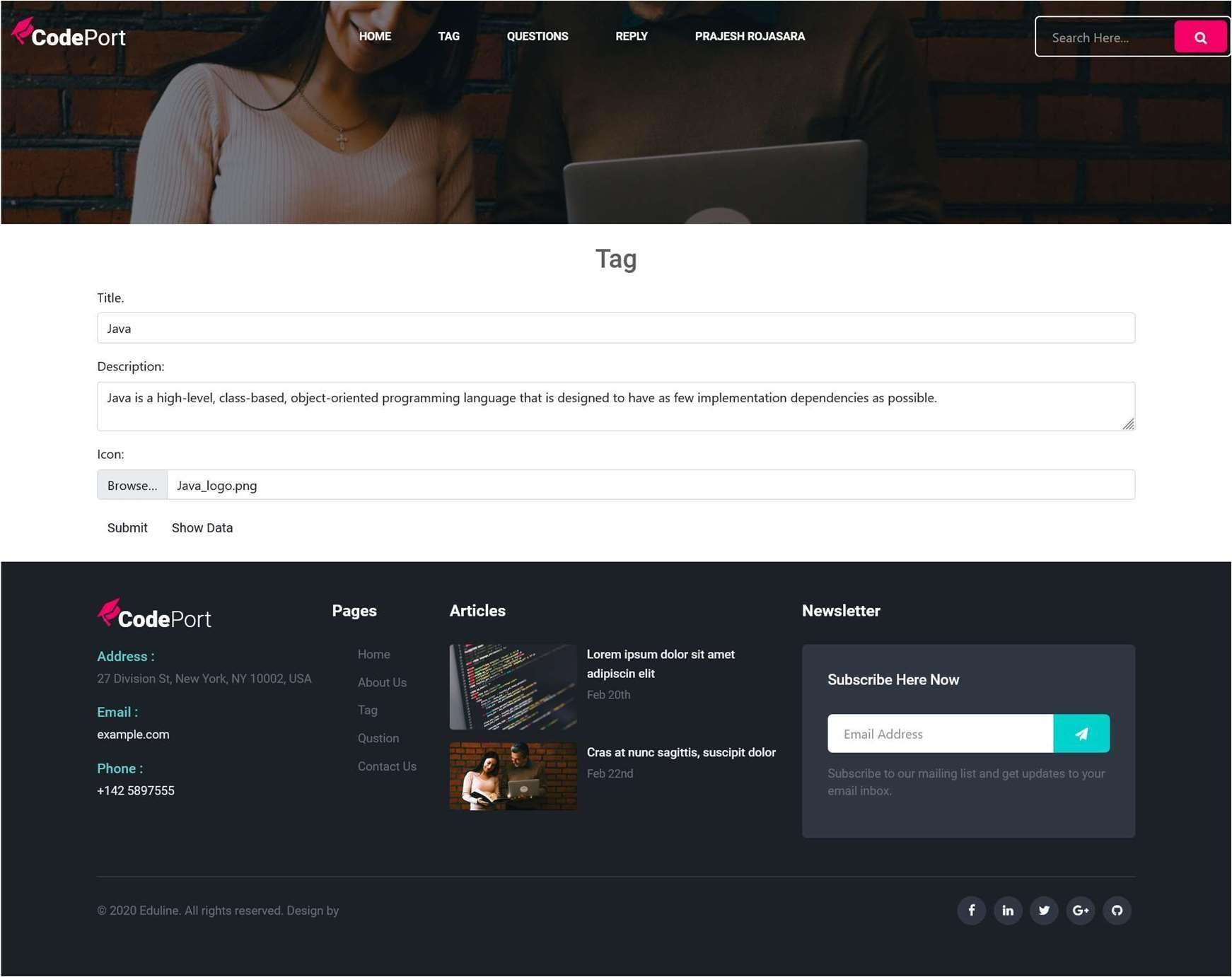
* This is user list page .
* Admin can give admin role to user.
* **Admin create new user page**:



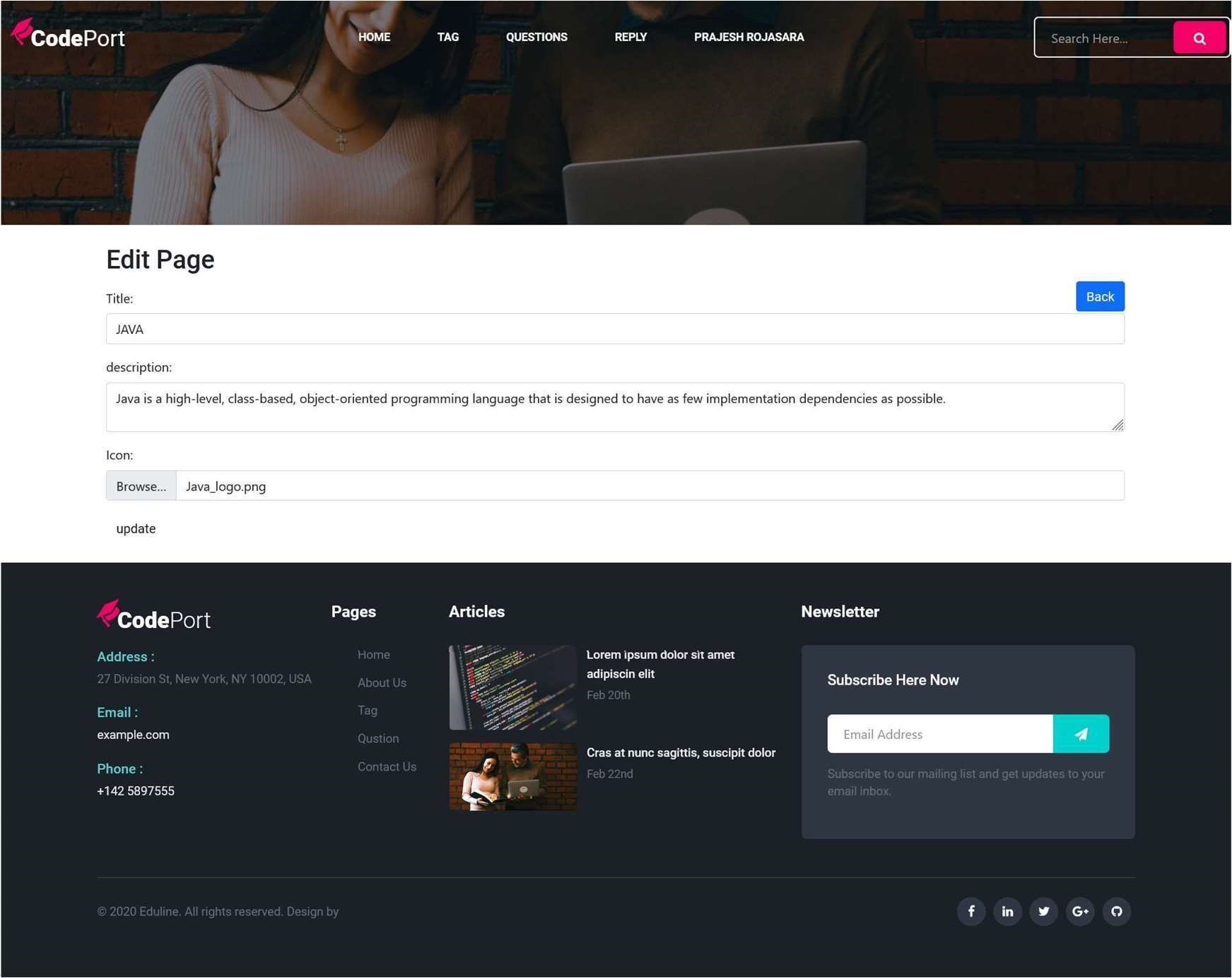
* Admin create new user with this page.
* Admin fill all details to create user.
* **Admin User Edit page:**



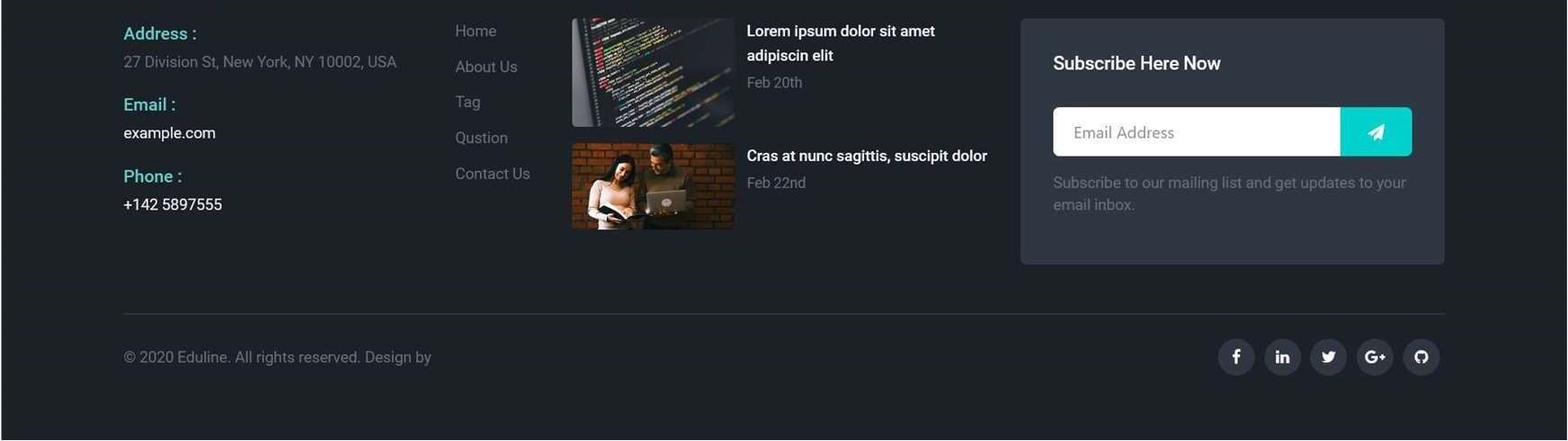
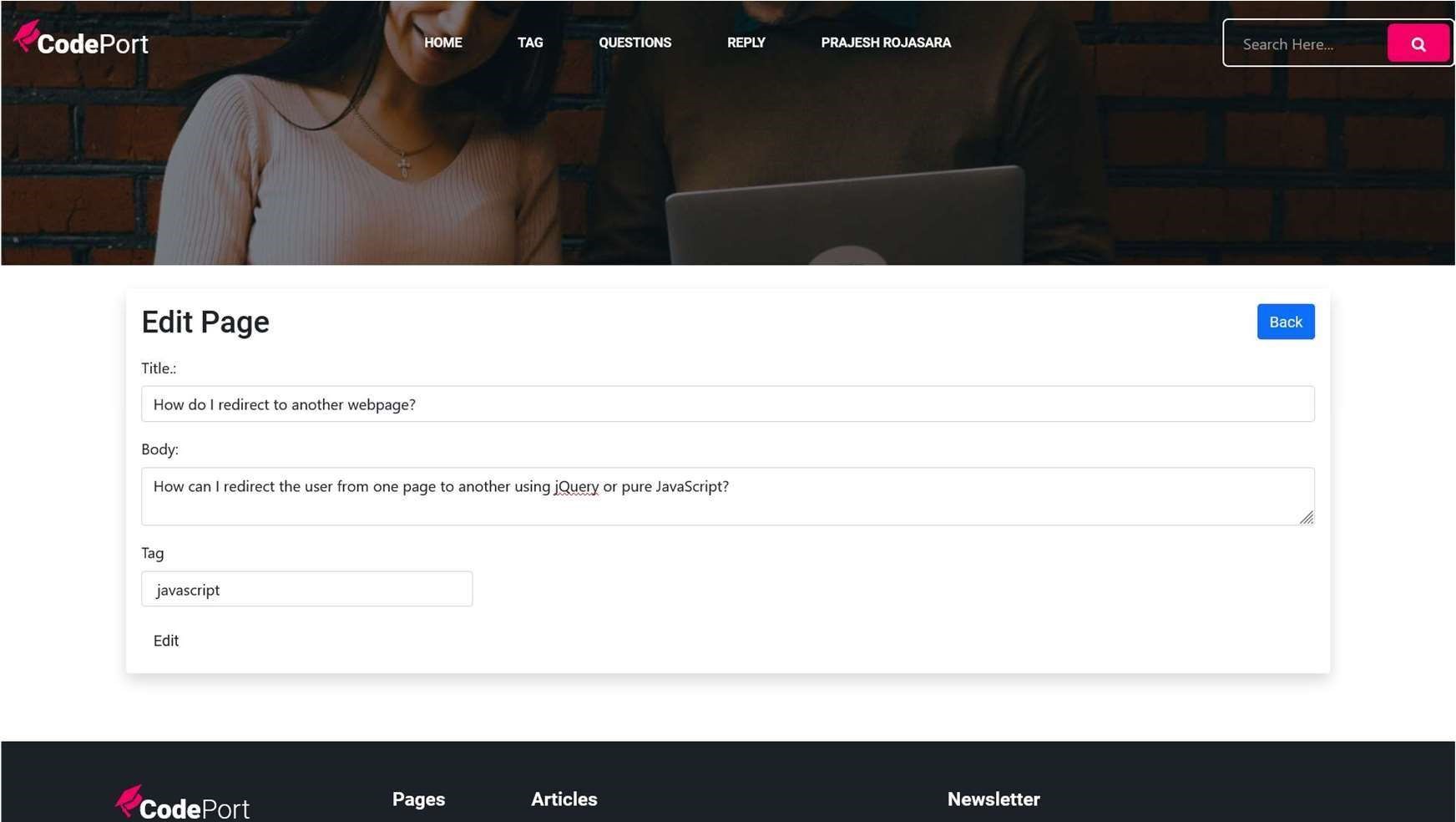
* This is edit page where User can edit his/her profile.
* **Admin add new Tag**



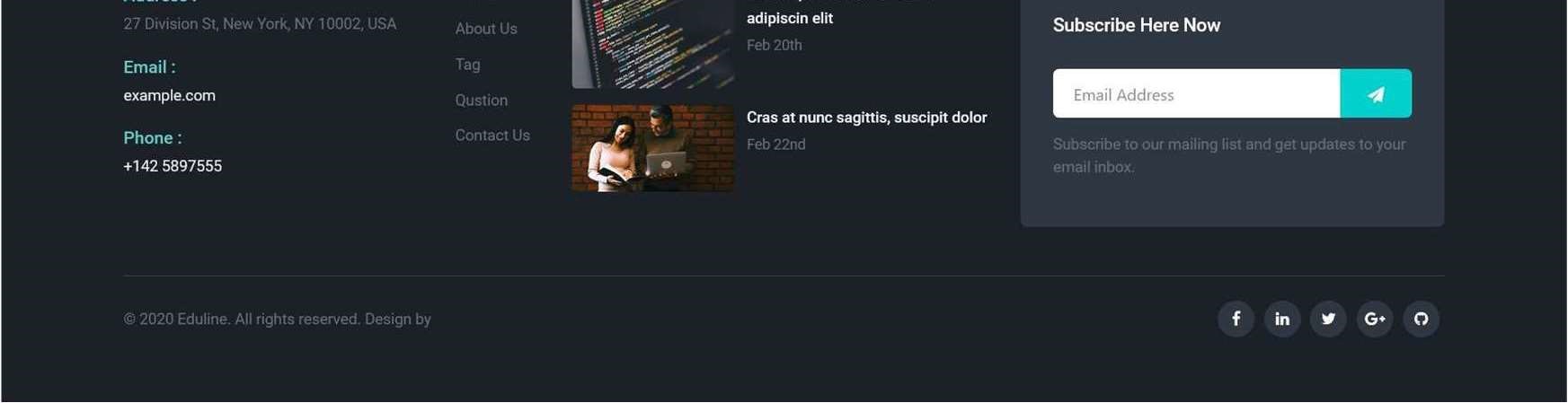
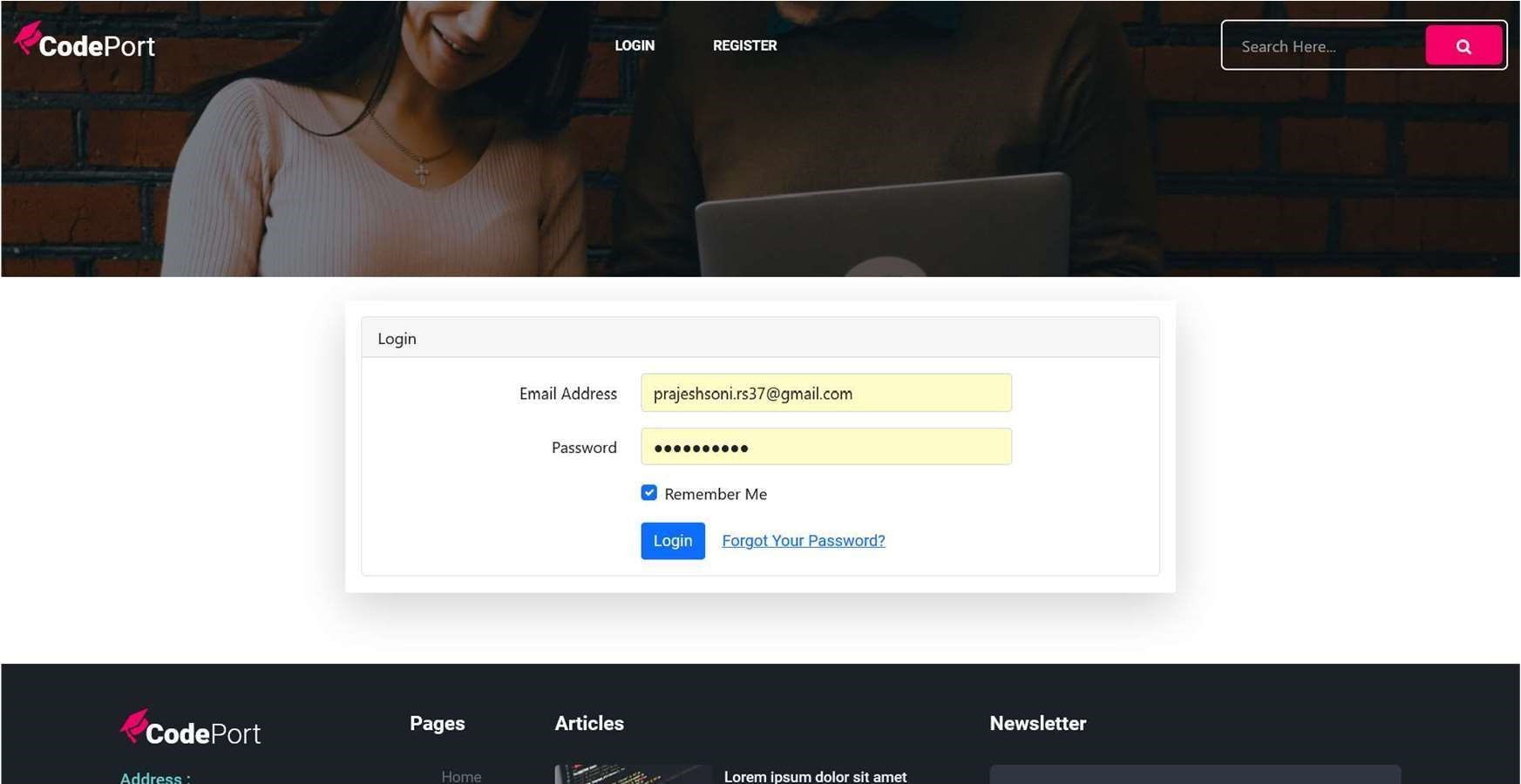
* Admin can add new tag.
* Admin can give tag title, description and icon
* **Admin Tag edit:**



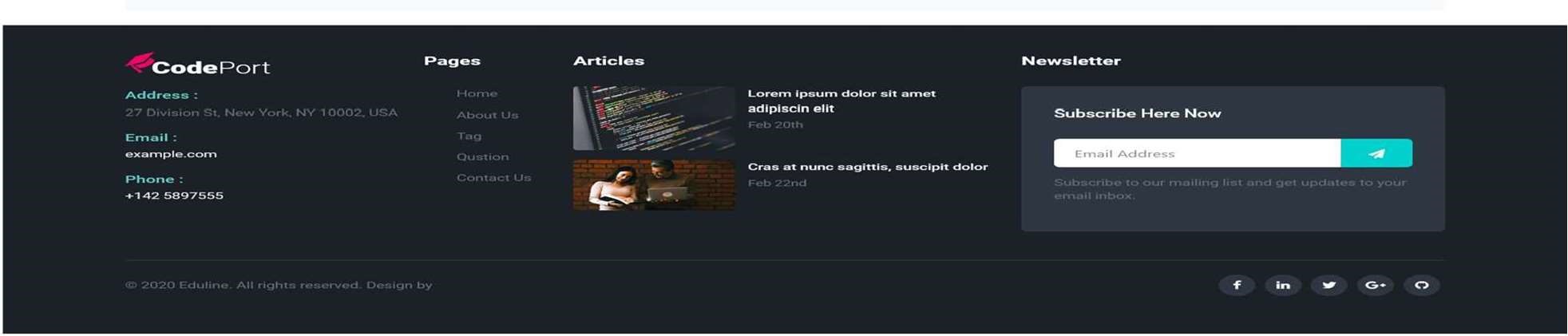
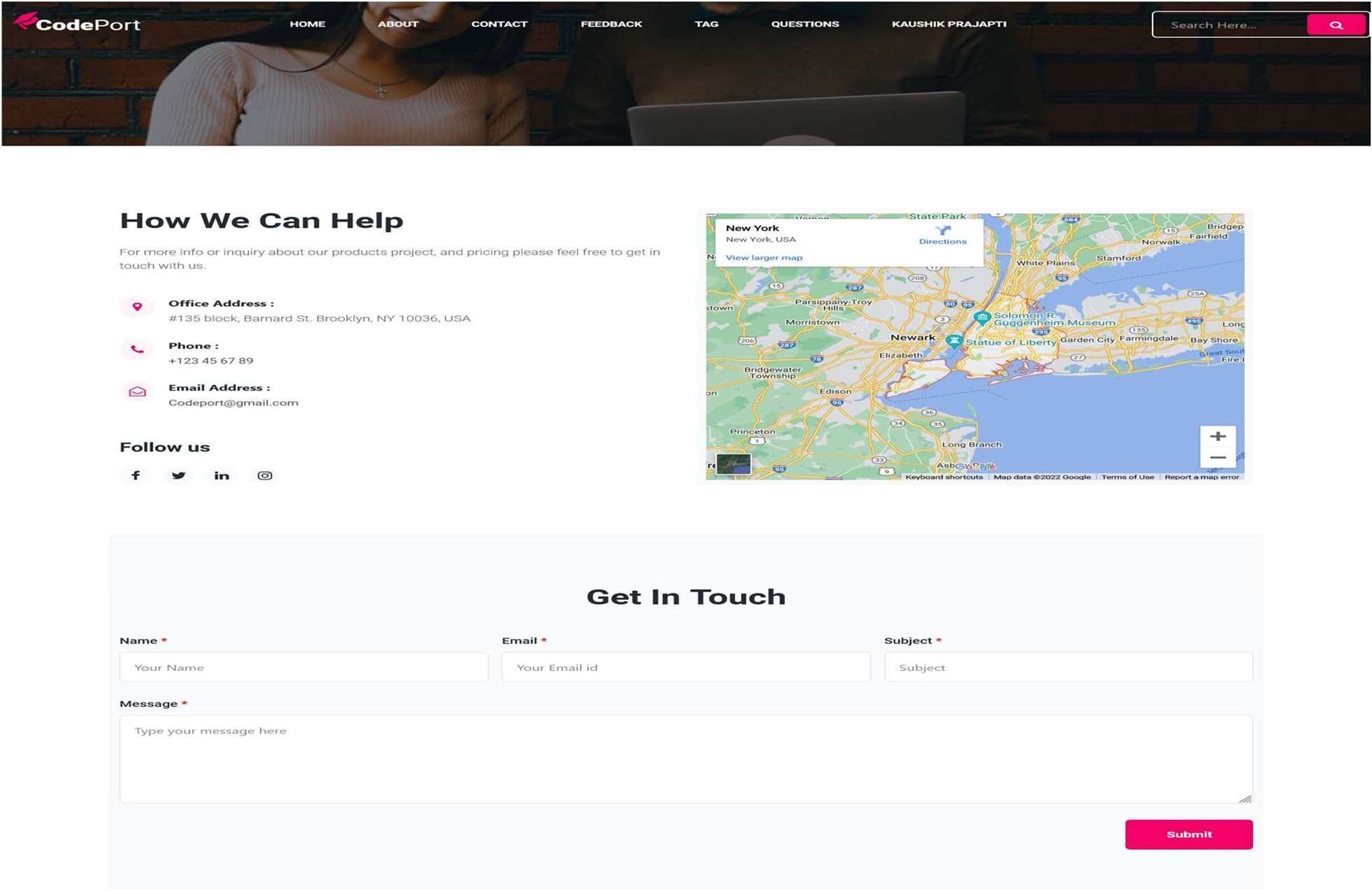
* This is edit page where admin can edit tag.
* **Admin Question Edit:**



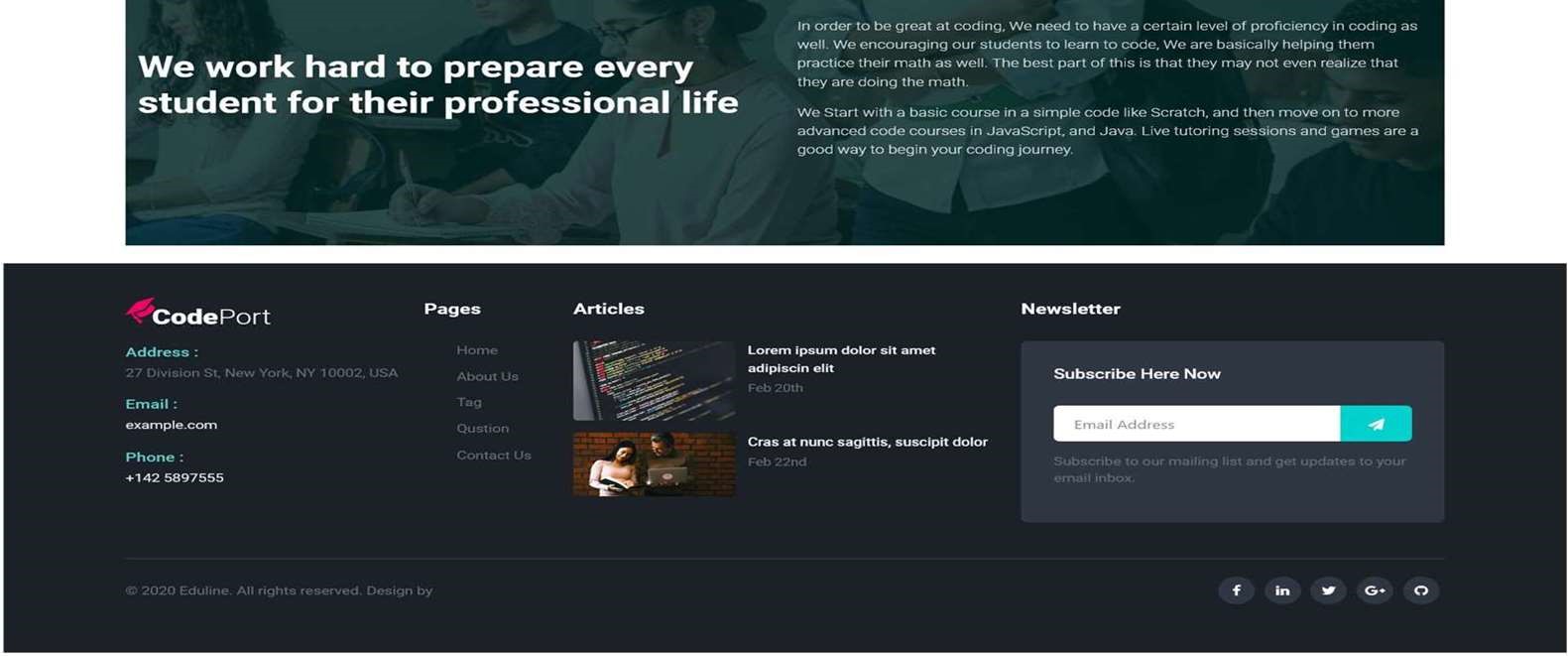
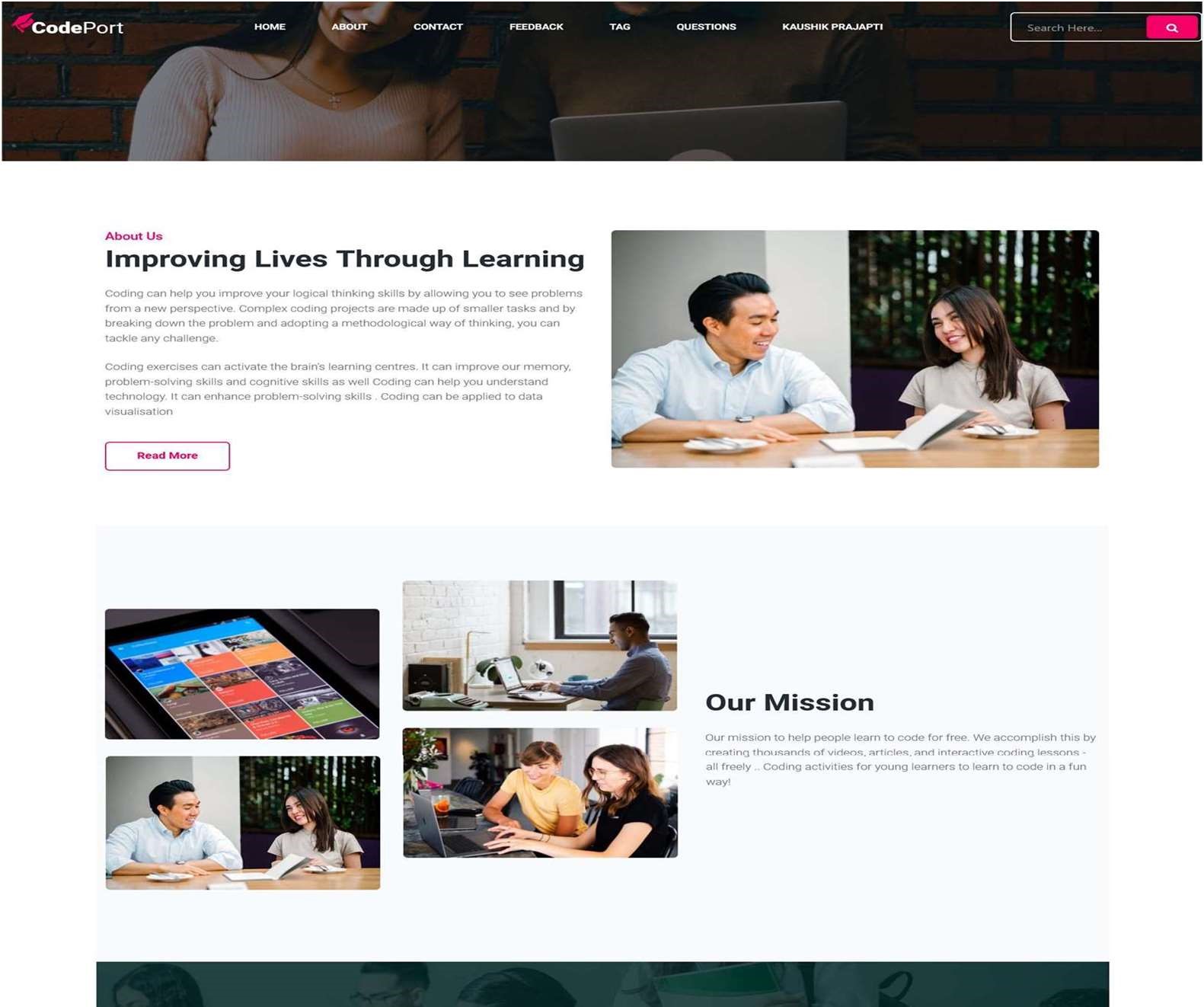
* This is question edit page where user can edit his/her question’s title, tag and body.
* **User Login:**



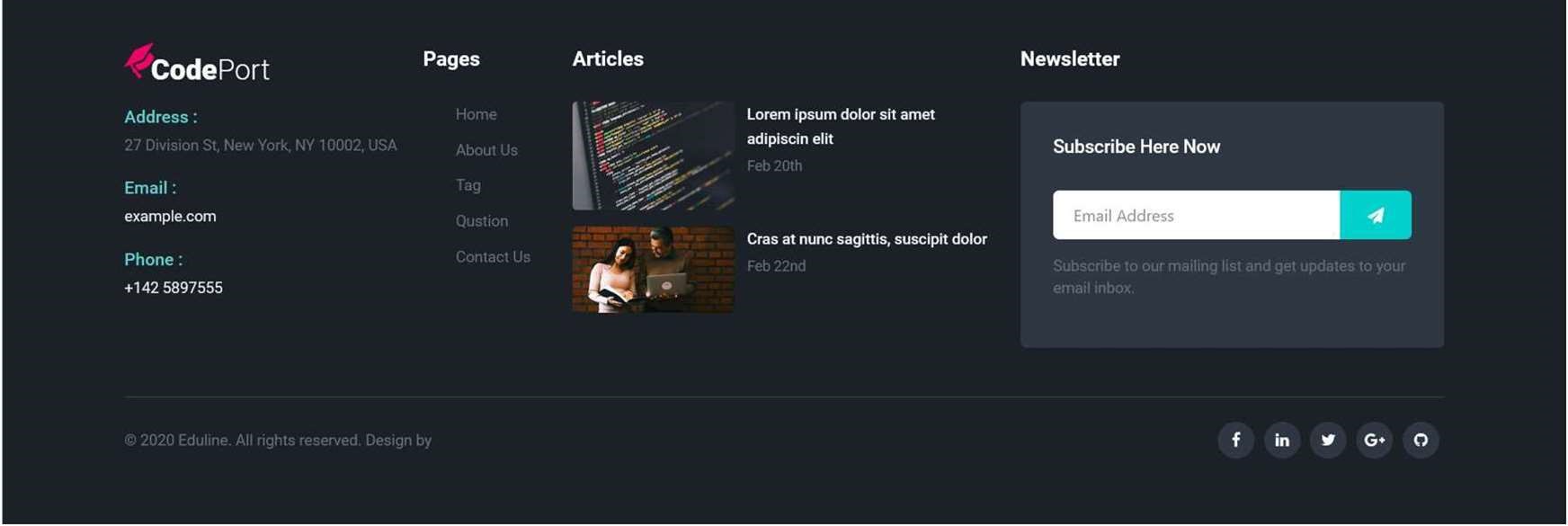
* Using this page user can login with his/her I’d and password.
* **User Contact page:**



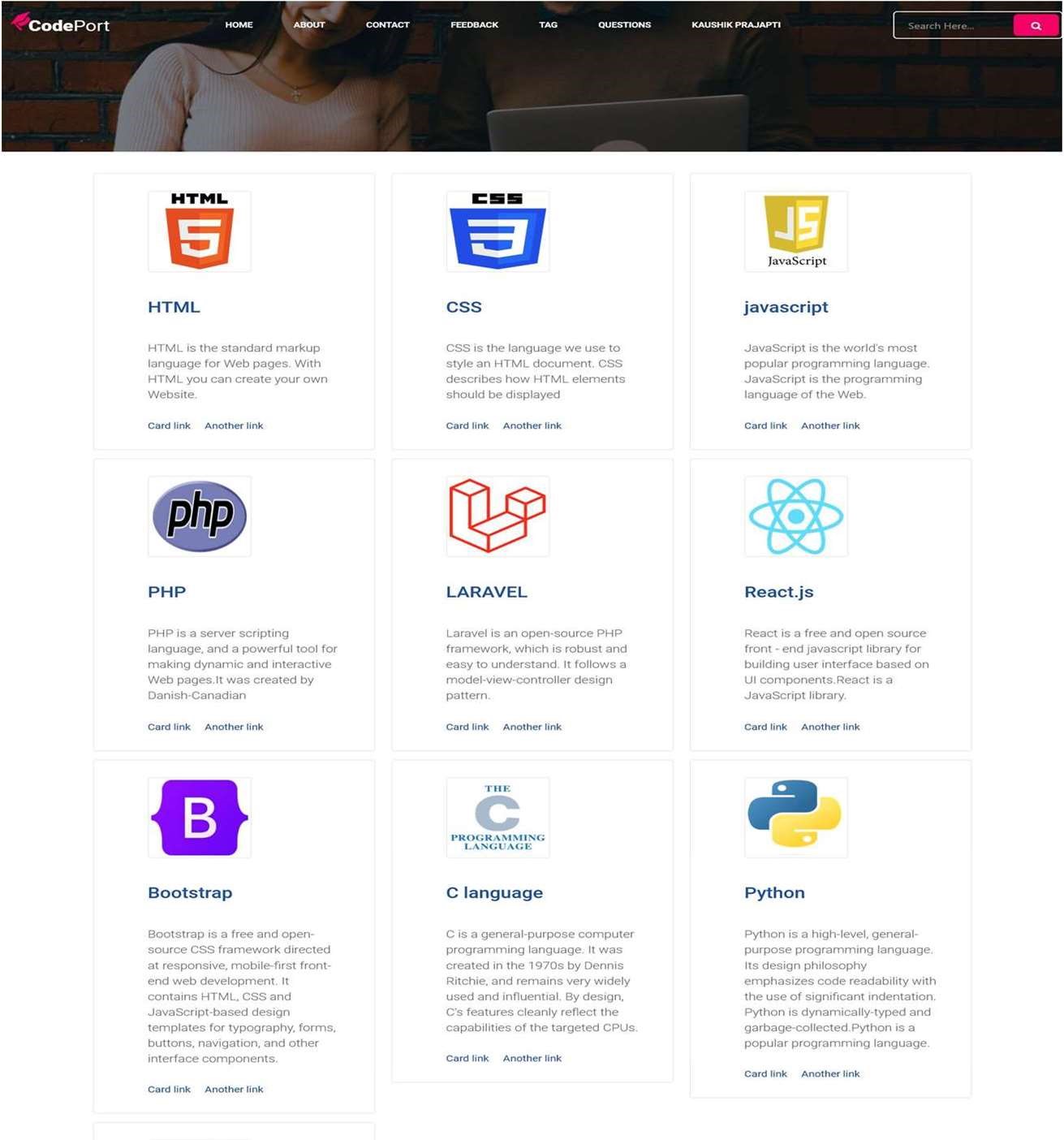
* Using contact page user can contact.
* **User About page:**



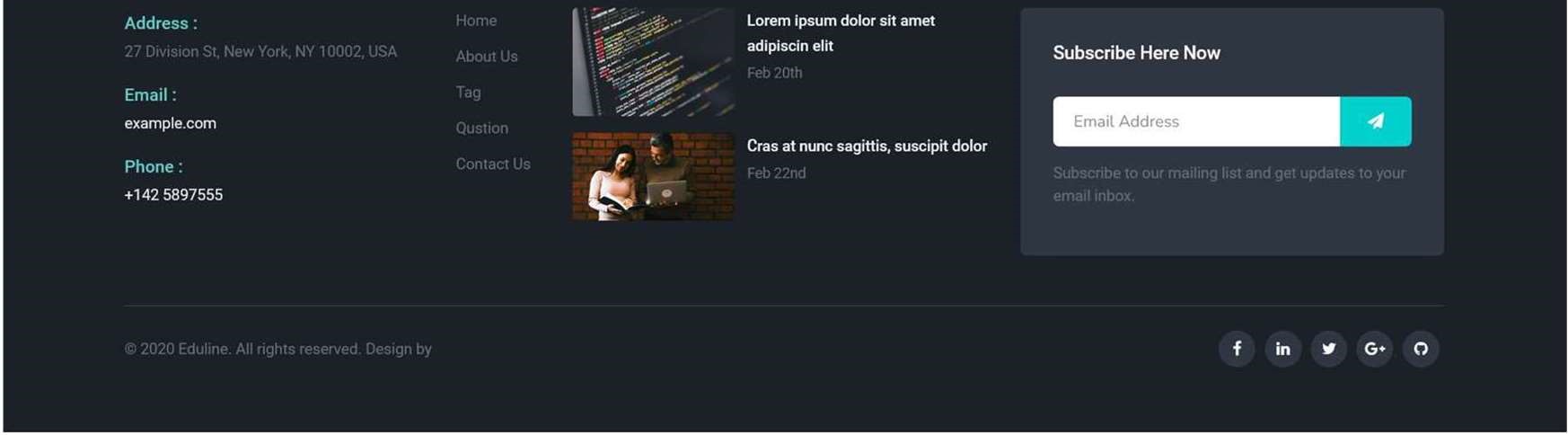
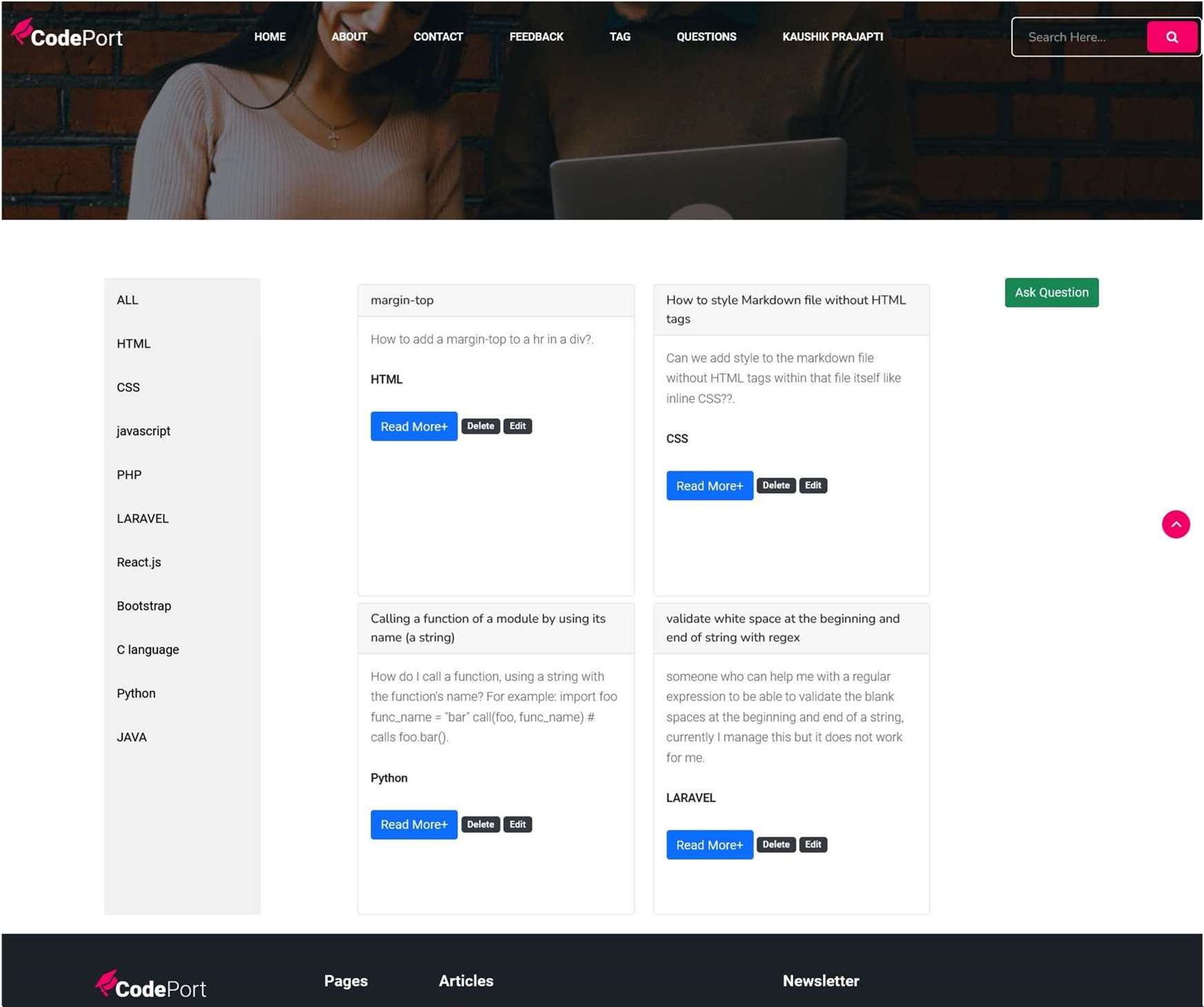
* This page is about admin.
* **User Feedback page:**



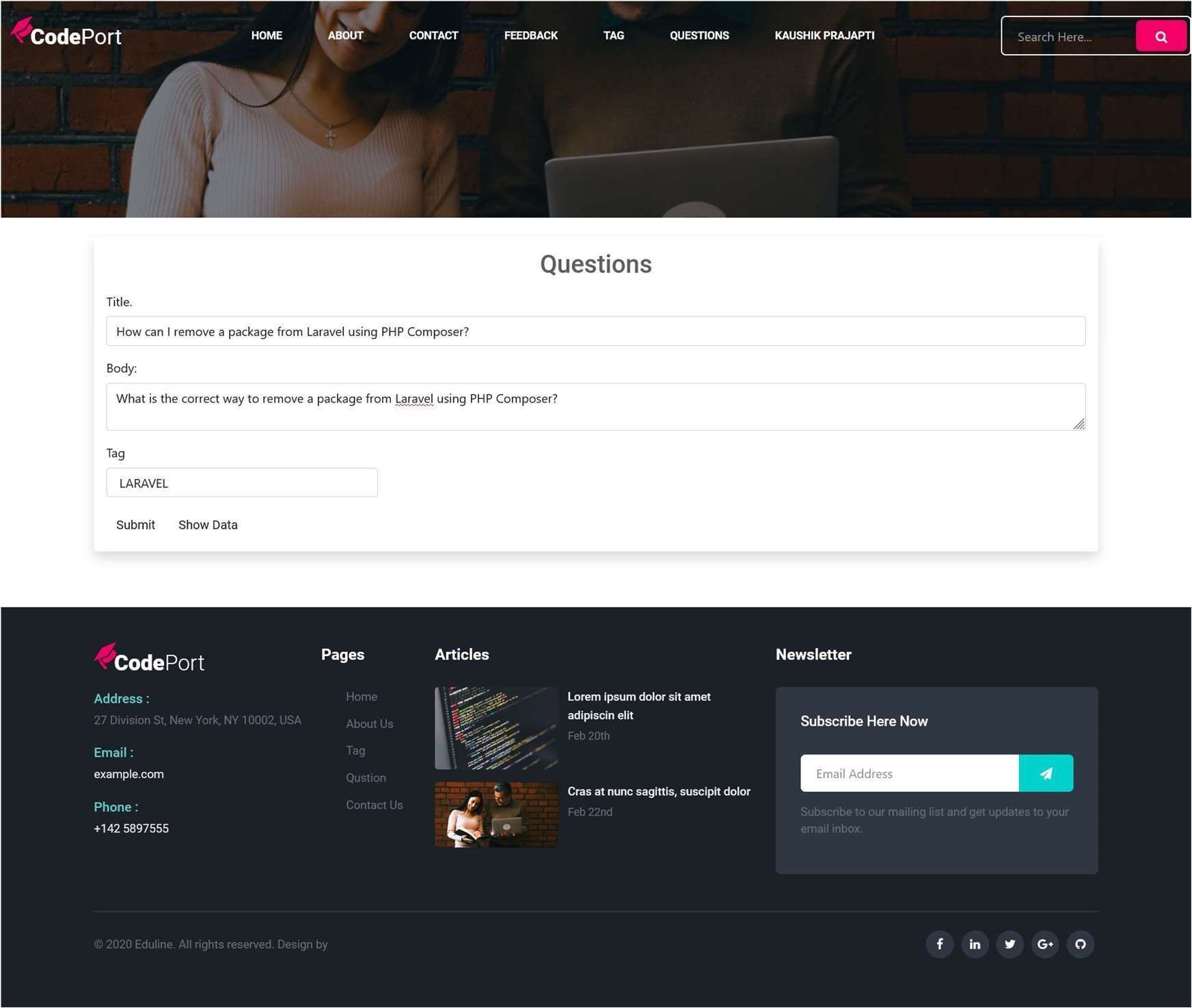
* User give feedback here .
* **User Tag page:**



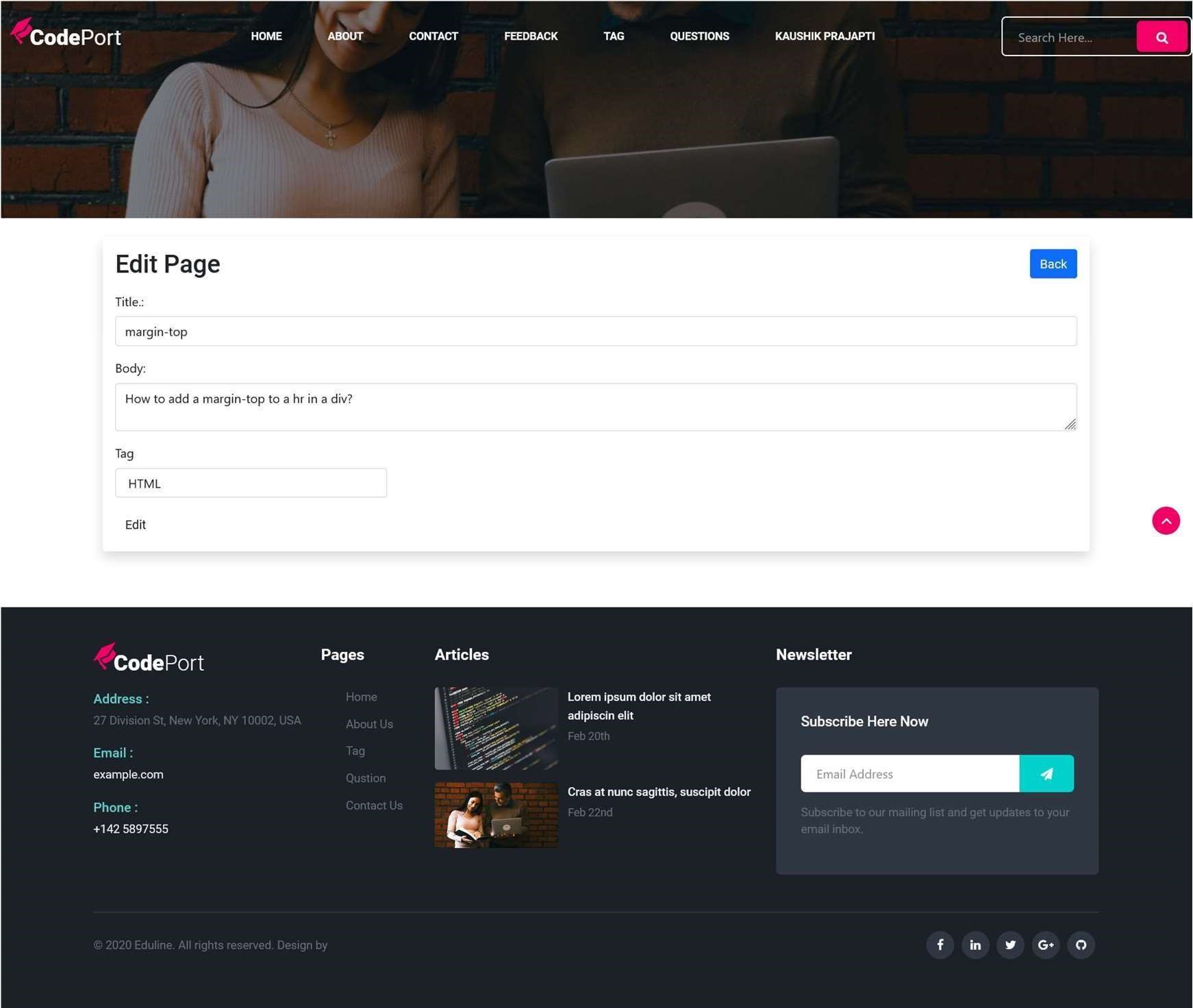
* This page is user tag page where all tags are describe.
* **User Question page:**



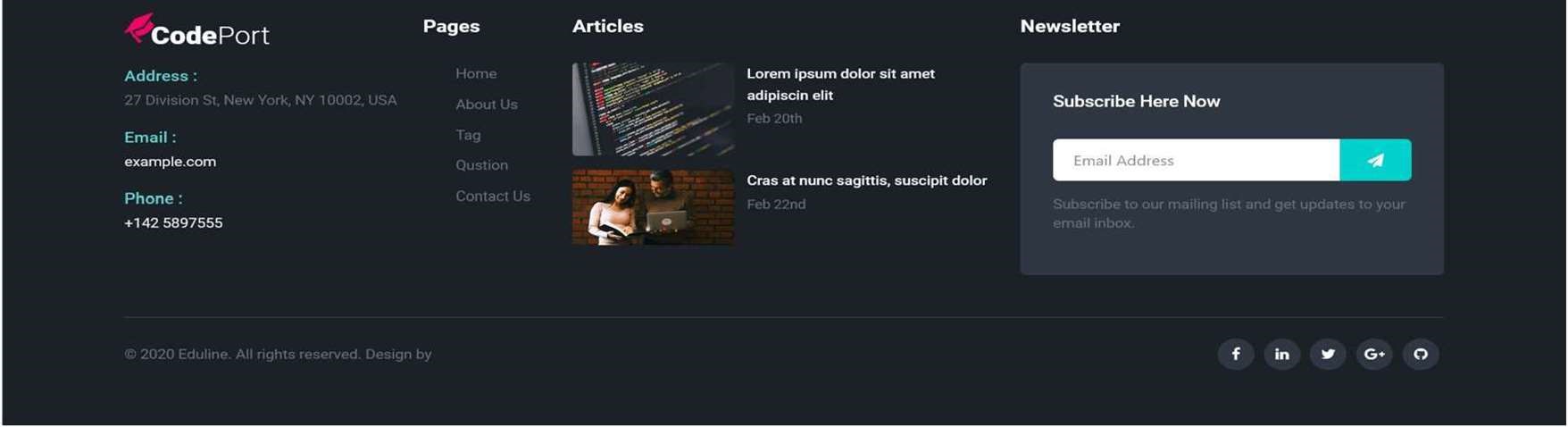
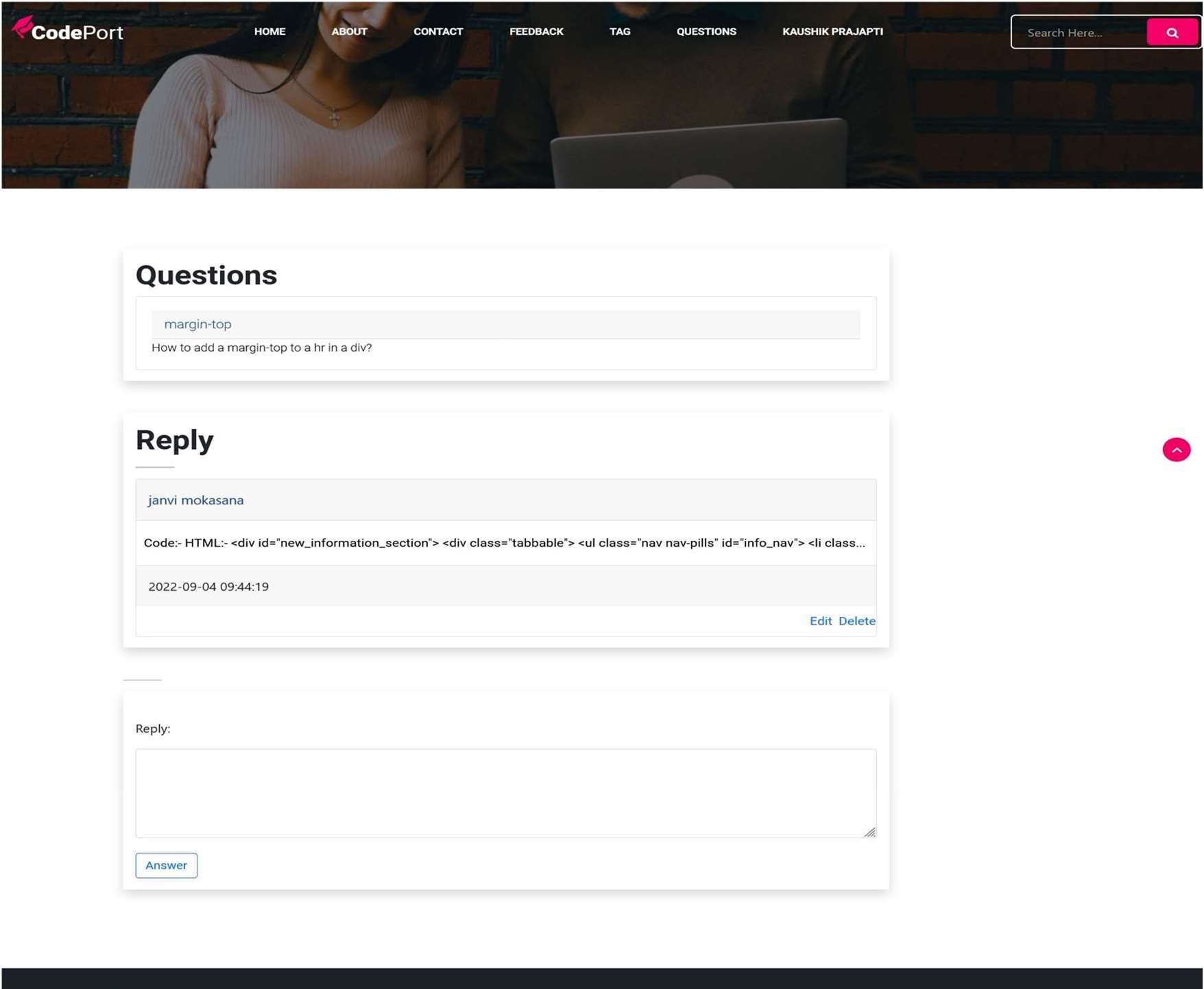
* This is user question page where user can ask his/her doubts about coding.
* **User Ask Question page:**



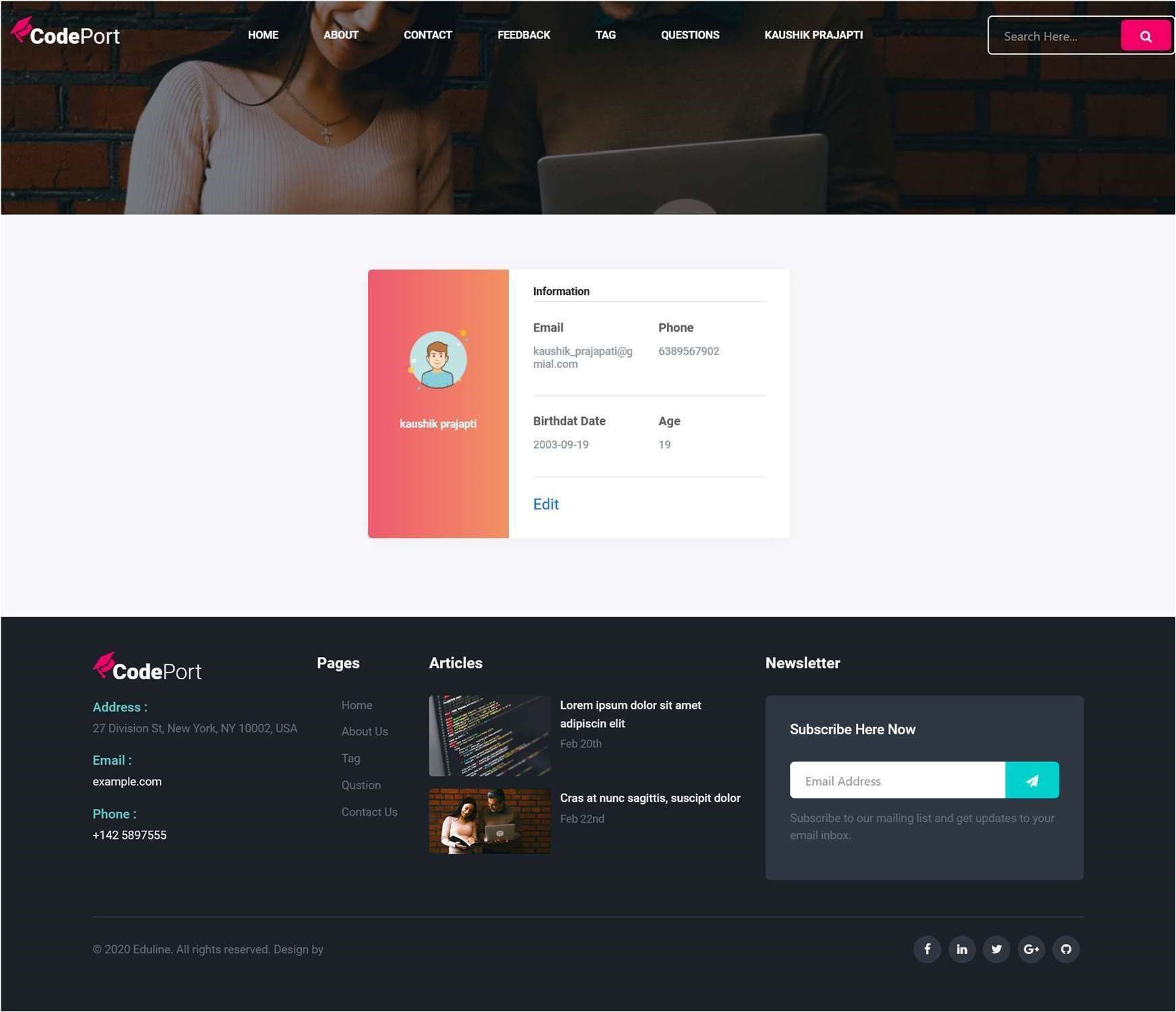
* User can ask his/her Question from this page.
* User must be give his/her question title, body and tag
* **User Edit Question page:**



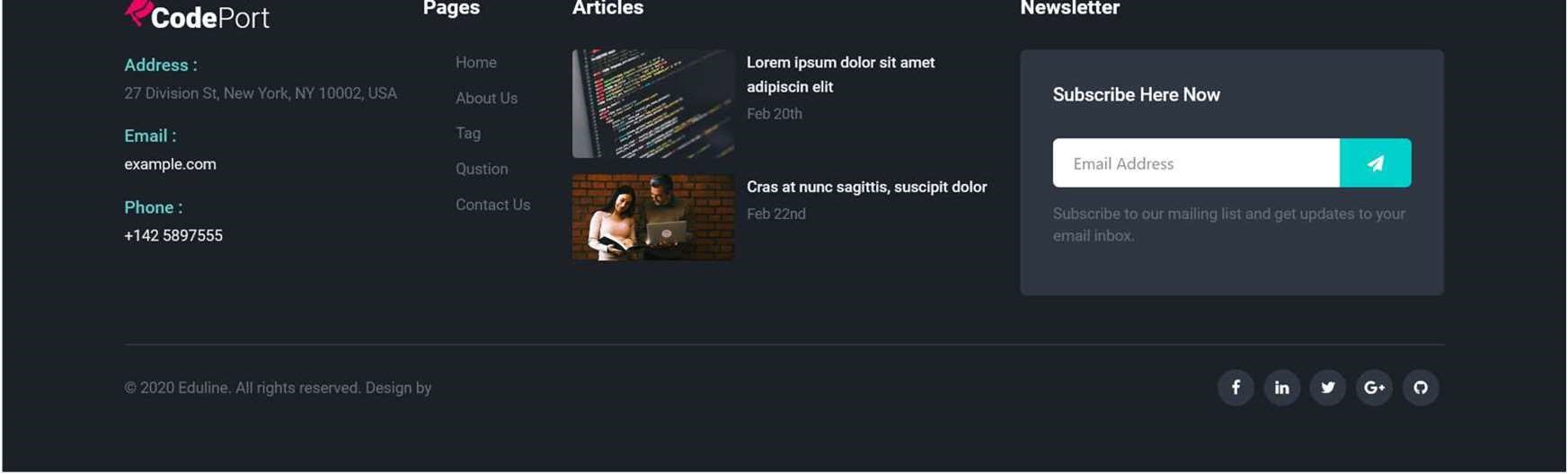
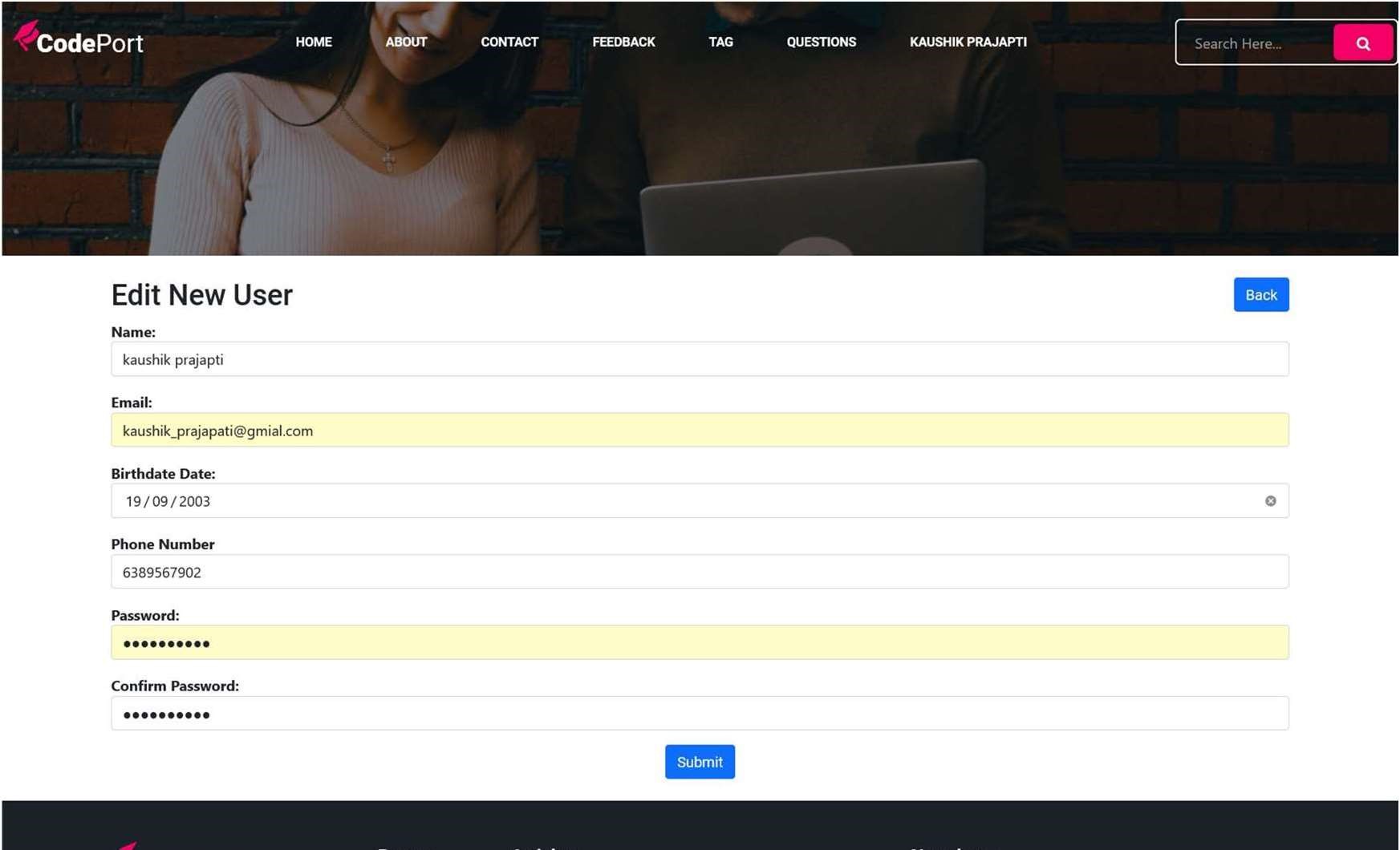
* User want edit his/her question’s body, title and tag then his/her edit from this page.
* **User Reply Page:**



* This is user reply page.
* Here user get their solution.
* **User Profile page:**



* Using this page user can see his/her profile.
* If user want to edit in profile they click edit button and edit profile.
* **User Edit page:**



* Here, user edit his/her profile.

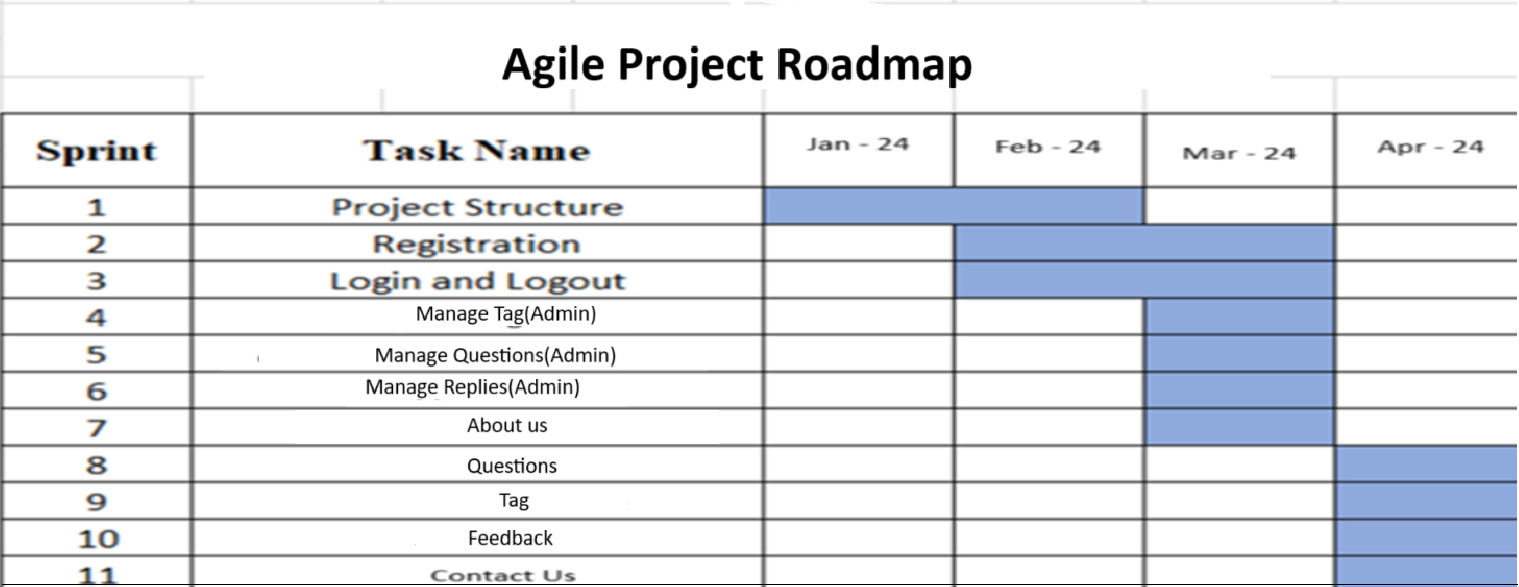
# **Agile Documentation**

# **Agile Project Charter:**

|  |  |
| --- | --- |
| General Project Information | |
| Project Name | Code Port Website |
| Project Champion | Kshama Gandhi, Purvi Makwana |
| Project Sponsor | LJ Institute of Computer Application |
| Project Manager | Prof. Jayshree Ma’am |
| Stakeholders | Admin , User |
| Expected Start Date | 10/01/2024 |
| Expected Completion Date | 05/04/2024 |

|  |  |
| --- | --- |
| Project Details | |
| Mission | Our mission is to create a vibrant and supportive online community where developers, programmers, and technology enthusiasts can collaborate, share knowledge, and learn from each other's experiences. |
| Vision | Our vision is to revolutionize the way developers and technology enthusiasts collaborate, learn, and grow together. |
| Scope | The scope of Code Port Website typically encompasses various features and functionalities aimed at replicating the core functionalities of the original platform. |
| Date | 10/04/2024 |

# **Agile Project Roadmap/Schedule:**



# **Agile Project Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** | **Status** |
| **Sprint#1 : Project Structure** | 7d | 10/01/2024 | 17/01/2024 | Completed |
| UI Designing | 4d | 17/01/2024 | 21/01/2024 | Completed |
| Database Management | 3d | 21/01/2024 | 24/01/2024 | Completed |
| **Sprint#2 : Registration** | 5d | 24/01/2024 | 29/01/2024 | Completed |
| User Registration | 5d | 29/01/2024 | 03/02/2024 | Completed |
| **Sprint#3 : Login and Logout** | 4d | 03/02/2024 | 07/02/2024 | Completed |
| User Login | 2d | 07/02/2024 | 09/02/2024 | Completed |
| Admin Login | 3d | 09/02/2024 | 12/02/2024 | Completed |
| **Sprint#4 : Manage Tag(Admin)** | 2d | 12/02/2024 | 14/02/2024 | Completed |
| Admin manages tag | 2d | 14/02/2024 | 16/02/2024 | Completed |
| **Sprint#5 : Manage Questions(Admin)** | 9d | 16/02/2024 | 25/02/2024 | Completed |
| Admin manages questions | 9d | 25/02/2024 | 05/03/2024 | Completed |
| **Sprint#6 : Manage Replies(Admin)** | 5d | 05/03/2024 | 10/03/2024 | Completed |
| Admin manages replies | 2d | 10/03/2024 | 12/03/2024 | Completed |
| **Sprint#7 : About Us** | 6d | 10/03/2024 | 16/03/2024 | Completed |
| View About Us Page | 2d | 16/03/2024 | 18/03/2024 | Completed |
| **Sprint#8 : Questions** | 1d | 18/03/2024 | 19/03/2024 | Completed |
| View and ask questions | 1d | 19/03/2024 | 20/03/2024 | Completed |
| **Sprint#9 : Tag** | 1d | 20/03/2024 | 21/03/2024 | Completed |
| View and select tag for the question | 1d | 21/03/2024 | 22/03/2024 | Completed |
| **Sprint#10 : Feedback** | 1d | 22/03/2024 | 23/03/2024 | Completed |
| Give Feedback | 1d | 23/03/2024 | 24/03/2024 | Completed |
| **Sprint#11 : Contact Us** | 4d | 24/03/2024 | 28/03/2024 | Completed |
| Contact us | 1d | 28/03/2024 | 29/03/2024 | Completed |

# **Agile User Story:**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID** | **As a (type of user)** | **I want to (perform some task)** | **So that I can (achieve some goal)** |
| **1** | Admin | Manage Login Module | Ensure that validate user can join |
| **2** | Admin | View User | View User |
| **3** | Admin | Manage password | Ensures the password is valid |
| **4** | User | About Us | Redirect to About Us page |
| **5** | User | Questions | Ask questions |
| **6** | User | Feedback | Give Feedback |
| **7** | User | Tag | Select tag for the questions |
| **8** | User | Contact Us | Redirect to Contact page |
| **9** | Admin | Manage Tag | Manage tag |
| **10** | Admin | Manage Questions and replies | Manage questions and replies |

# **Agile Release Plan :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** | **Status** |
| **Sprint#1 : Project Structure** | 7d | 10/01/2024 | 17/01/2024 | Completed |
| UI Designing | 4d | 17/01/2024 | 21/01/2024 | Completed |
| Database Management | 3d | 21/01/2024 | 24/01/2024 | Completed |
| **Sprint#2 : Registration** | 5d | 24/01/2024 | 29/01/2024 | Completed |
| User Registration | 5d | 29/01/2024 | 03/02/2024 | Completed |
| **Sprint#3 : Login and Logout** | 4d | 03/02/2024 | 07/02/2024 | Completed |
| User Login | 2d | 07/02/2024 | 09/02/2024 | Completed |
| Admin Login | 3d | 09/02/2024 | 12/02/2024 | Completed |
| **Sprint#4 : Manage Tag(Admin)** | 2d | 12/02/2024 | 14/02/2024 | Completed |
| Admin manages tag | 2d | 14/02/2024 | 16/02/2024 | Completed |
| **Sprint#5 : Manage Questions(Admin)** | 9d | 16/02/2024 | 25/02/2024 | Completed |
| Admin manages questions | 9d | 25/02/2024 | 05/03/2024 | Completed |
| **Sprint#6 : Manage Replies(Admin)** | 5d | 05/03/2024 | 10/03/2024 | Completed |
| Admin manages replies | 2d | 10/03/2024 | 12/03/2024 | Completed |
| **Sprint#7 : About Us** | 6d | 10/03/2024 | 16/03/2024 | Completed |
| View About Us Page | 2d | 16/03/2024 | 18/03/2024 | Completed |
| **Sprint#8 : Questions** | 1d | 18/03/2024 | 19/03/2024 | Completed |
| View and ask questions | 1d | 19/03/2024 | 20/03/2024 | Completed |
| **Sprint#9 : Tag** | 1d | 20/03/2024 | 21/03/2024 | Completed |
| View and select tag for the question | 1d | 21/03/2024 | 22/03/2024 | Completed |
| **Sprint#10 : Feedback** | 1d | 22/03/2024 | 23/03/2024 | Completed |
| Give Feedback | 1d | 23/03/2024 | 24/03/2024 | Completed |
| **Sprint#11 : Contact Us** | 4d | 24/03/2024 | 28/03/2024 | Completed |
| Contact us | 1d | 28/03/2024 | 29/03/2024 | Completed |

# **Agile Sprint Backlog:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** | **Status** |
| **Sprint#1 : Project Structure** | 7d | 10/01/2024 | 17/01/2024 | Completed |
| UI Designing | 4d | 17/01/2024 | 21/01/2024 | Completed |
| Database Management | 3d | 21/01/2024 | 24/01/2024 | Completed |
| **Sprint#2 : Registration** | 5d | 24/01/2024 | 29/01/2024 | Completed |
| User Registration | 5d | 29/01/2024 | 03/02/2024 | Completed |
| **Sprint#3 : Login and Logout** | 4d | 03/02/2024 | 07/02/2024 | Completed |
| User Login | 2d | 07/02/2024 | 09/02/2024 | Completed |
| Admin Login | 3d | 09/02/2024 | 12/02/2024 | Completed |
| **Sprint#4 : Manage Tag(Admin)** | 2d | 12/02/2024 | 14/02/2024 | Completed |
| Admin manages tag | 2d | 14/02/2024 | 16/02/2024 | Completed |
| **Sprint#5 : Manage Questions(Admin)** | 9d | 16/02/2024 | 25/02/2024 | Completed |
| Admin manages questions | 9d | 25/02/2024 | 05/03/2024 | Completed |
| **Sprint#6 : Manage Replies(Admin)** | 5d | 05/03/2024 | 10/03/2024 | Completed |
| Admin manages replies | 2d | 10/03/2024 | 12/03/2024 | Completed |
| **Sprint#7 : About Us** | 6d | 10/03/2024 | 16/03/2024 | Completed |
| View About Us Page | 2d | 16/03/2024 | 18/03/2024 | Completed |
| **Sprint#8 : Questions** | 1d | 18/03/2024 | 19/03/2024 | Completed |
| View and ask questions | 1d | 19/03/2024 | 20/03/2024 | Completed |
| **Sprint#9 : Tag** | 1d | 20/03/2024 | 21/03/2024 | Completed |
| View and select tag for the question | 1d | 21/03/2024 | 22/03/2024 | Completed |
| **Sprint#10 : Feedback** | 1d | 22/03/2024 | 23/03/2024 | Completed |
| Give Feedback | 1d | 23/03/2024 | 24/03/2024 | Completed |
| **Sprint#11 : Contact Us** | 4d | 24/03/2024 | 28/03/2024 | Completed |
| Contact us | 1d | 28/03/2024 | 29/03/2024 | Completed |

# **Agile Test Plan:**

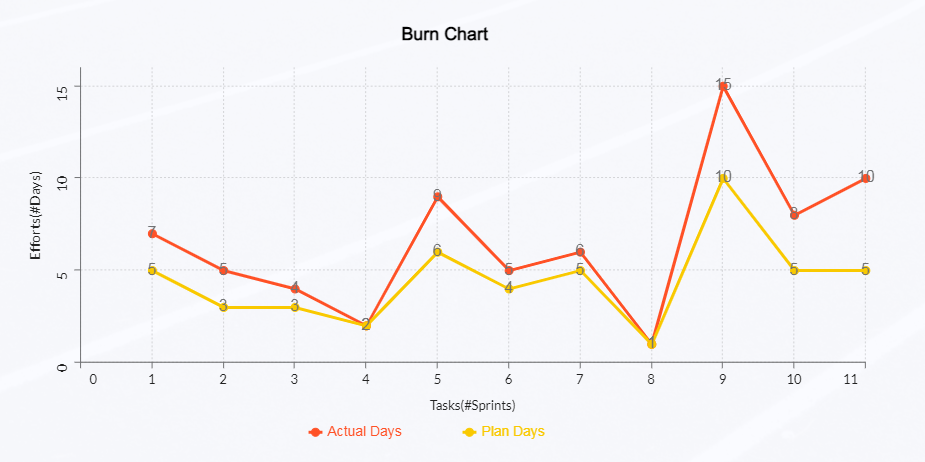
|  |  |
| --- | --- |
| **Test Case ID : 101** | **Test Case Name : User Login** |
| **Designed by** : **Kshama Gandhi** | **Design Date :- 05/02/2024** |
| **Module Name : User Login** | **Sub Module Name : Login** |
| **Executed by : Kshama Gandhi** | **Execution Date : 05/02/2024** |
| **Brief Description : User Login**  **(with Data : “email : t@g.com & password:1234”)** | **Test Priority (Low/Medium/High) : High** |
| **Pre-Conditions: User Must Registered & have** | **Correct Username & Password.** |
| **Dependencies : None** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Remarks** |
| 1 | **Email:** t@g.com  **Password:** 1234 | Successfully login & redirect to  Home Page. | Login Successful. | Pass | None |
| 2 | **Email:** t1@g.com  **Password:** 1234 | Show “Invalid Credentials.” | Login Not Successful | Fail | Need to Show Error  Message. |
| 3 | **Email:** t@g.com  **Password:** 1234Mah | Show “Invalid Credentials.” | Login Not Successful | Fail | Need to Show Error  Message. |
| 4 | **Email:** t@g.com  **Password:** 1234 | Successfully login in to system &  redirect to home Page. | Login Successful & Redirect to home Page. | Pass | None |

|  |  |
| --- | --- |
| **Test Case ID : 102** | **Test Case Name : User Registration** |
| **Designed by : Purvi Makwana** | **Design Date : 30/03/2024** |
| **Module Name : User Registration** | **Sub Module Name : Register** |
| **Executed by : Purvi Makwana** | **Execution Date : 30/03/2024** |
| **Brief Description: User Registration page for the new user. User needs to provide**  **the correct details for Registration.** | **Test Priority (Low/Medium/High) : High** |
| **Pre-conditions : None** | |
| **Dependencies: Username & Password are store & fetch from auth\_user Table.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Remarks** |
| 1 | **First Name:** Purvi **Last Name:** Makwana  **Email:** [purvi@gmail.com](mailto:purvi@gmail.com) **Password:** 1234 **Confirm Password:**  1234 | Successfully Registered & Redirect to Login Page. | Registered Successfully. | Pass | redirect to login Page. |
| 2 | **First Name:** Purvi **Last Name:** Makwana **Email:** [purvi@gmail.com](mailto:purvi@gmail.com) **Password:** 1234 **Confirm Password:**  12345 | Show Error Message as “Password & Confirm  Password must be Equal” | Not Registered & remain in Registration page. | Fail | None |
| 3 | **First Name:** Purvi **Last Name:** Makwana **Email:** [purvi@gmail.com](mailto:purvi@gmail.com) **Password:** 1234 **Confirm Password:**  1234 | Show Error Message as “Email  Id Already Exist” | Not Registered & remain in Registration page. | Fail | None |
| 4 | **First Name:** Purvi **Last Name:** Makwana **Email:** [purvi@gmail.com](mailto:purvi@gmail.com) **Password:** 4321 **Confirm Password:**  4321 | Successfully Registered & | Successfully Registered & | Pass | None |

# **Earned-Value and Burn Chart:**



# **Proposed Enhancements:**

To enhance Code Port Website built in Laravel, several features and improvements can be proposed. Firstly, implementing a more robust user reputation system could be beneficial, allowing users to gain reputation points through various activities such as asking insightful questions, providing helpful answers, and contributing to the community in other ways. This reputation system could also include badges or achievements to further incentivize positive participation.

Secondly, improving the search functionality could greatly enhance user experience. Implementing advanced search filters, including options to search by tags, user reputation, and date ranges, can help users quickly find relevant content. Additionally, incorporating a suggestion feature that provides autocomplete suggestions as users type their queries can streamline the search process.

Thirdly, enhancing the notification system can help users stay updated on relevant activities within the platform. This could include real-time notifications for responses to their questions or comments, new answers to questions they follow, or updates on topics they are interested in.

# **Conclusion:**

In conclusion, a database is a far more efficient mechanism to store and organize data than spreadsheets. It allows for a centralized facility that can easily be modified and quickly shared among multiple users. Having a web-based front end removes the requirement of users having to understand and use a database directly, and allows users to connect from anywhere with an internet connection and a basic web browser. It also allows the possibility of queries to obtain information for various surveys. This application is used for keeping track of employee information. It is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small-scale organizations where the number of employees is limited. This database consists of different entities like the profile details, application for leave, and pay slip. According to the requested requirement, the admin can add, manipulate, update and delete all employee data in his organization. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based every month. Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost-effective and manage records efficiently.

# **Bibliography:**

* <https://reactjs.org/docs/getting-started.html>
* <https://stackoverflow.com>
* <https://www.youtube.com>
* <https://app.diagrams.net>
* <https://nodejs.org/en/docs>
* <https://expressjs.com/>
* <https://www.mongodb.com>