Software Requirement Specification

(SRS)

Head Start Web Application

Ahmad Mohammad Al-Nsour

Date: 09-01-2020

Version 1.0.0

Table of Contents

1. INTRODUCTION

The purpose of Head Start is to outline the requirements for The Coding Academy's by Orange With interactive, unique and unusual ideas. Head Start will be built on ReactJS, Express, and MongoDB with Bootstrap and Font Awesome. It will be Web Application and accessible with any standard compliant browser. This guide will cover understanding the overview of the website, how it works and the technology behind it.

1.1 PURPOSE

- ✓ Increase the chances in getting a job.
- ✓ Stay up-to-date with all new interview HR/Technical questions.

1.2 INTENDED AUDIENCE AND PERTINENT SECTIONS

The intended audience for Head Start is web designers, people who are interested in HR and Technical interview questions and any related events & activities, students and faculties who are currently using or are interested in using the web platform.

Stockholder's are:

- ✓ Users.
- ✓ Guests.
- ✓ HR/Technical Admins.
- ✓ Owner.

1.3 PROJECT SCOPE

The Head Start website is an application. The purpose of the website is to resolve the client to allow website users to perform tasks related to HR and Technical questions and related events. Non-member users are only allowed to see Landing, HR questions, Technical question, Events and About Us page; non-member users are required to create an account in order to see the questions answer. Member users are required to login into their account prior to view questions answer (FaceBook, 2013), they have the right to AskQuestion, JoinUs, and add comments and to edit their profile information, Member with Admin role will be able to add a new question and answer question depends on their field.

- ✓ This Project will be used to provide users with the most frequent interview HR and Technical Questions.

- ✓ Allow users to take an online Quiz to evaluate themselves.

1.4 DOCUMENT CONVENTIONS

- JS: JavaScript
- UI: User Interface.
- SQL: Structured Query Language.
- CSS: Cascading Style Sheets.
- Mern Stack: MongoDB, ExpressJS, ReactJS, NodeJS.
- OS: Operating system.
- Git: A distributed version-control system for tracking changes in source code during software development.
- GitHub: A global company that provides hosting for software development version control using Git.
- Framework: A general “blank” application that can be easily customized into a wide variety of specific real applications. A framework provides code that establishes an ideal complete application that knows “what” to do, but it doesn't know “how” to do it.
- Library: A library is a collection of precompiled routines that a program can use. The routines, sometimes called modules, are stored in object format. Libraries are particularly useful for storing frequently used routines because you do not need to explicitly link them to every program that uses them.
- MongoDB: A document-oriented, No-SQL database used to store the application data.
- ExpressJS: A framework layered on top of NodeJS, used to build the backend of a site using NodeJS functions and structures. Since NodeJS was not developed to make websites but rather run JavaScript on a machine, ExpressJS was developed.
- ReactJS: A library created by Facebook. It is used to build UI components that create the user interface of the single page web application.
- NodeJS: The JavaScript runtime environment. It is used to run JavaScript on a machine rather than in a browser.
- Axios: A promise based HTTP client for the browser and Node.js. Axios makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations.

- Firebase: A Backend-as-a-Service — baaS — that started as an YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform.
- UML: Short for Unified Modeling Language is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.
- Use Case: A list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an *actor*) and a system to achieve a goal.
- Schema: structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases).

1.5 REFERENCES

Bert Bos, H. W. (1996, December 17). *CSS*.
Retrieved from https://www.w3schools.com/css/default.asp
Dahl, R. (2009). *NodeJS*.
Retrieved from https://nodejs.org/en/
Dwight Merriman, E. H. (2018). *MongoDB*.
Retrieved from https://www.mongodb.com/
FaceBook. (2013). *ReactJS*.
Retrieved from https://reactjs.org/
FaceBook. (n.d.). *React-Router-dom*.
Retrieved from https://reacttraining.com/react-router/web/guides/quick-start
Gandy, D. (2019, December 10). *Font awesome*.
Retrieved from https://fontawesome.com/
Holowaychuk, T. (2009, May 27). *ExpressJS*.
Retrieved from https://expressjs.com/
Lee, J. T. (2012, April 12). *Firebase*.
Retrieved from https://firebase.google.com/
Rebecca Turner, K. M. (2010, January 12). *Npm*.
Retrieved from https://docs.npmjs.com/
Scott Chacon, T. P.-W. (2008, February 8). *GitHub*.
Retrieved from https://github.com/
Simsek, G. (2017, FEBRUARY 02). *Robo3t*.
Retrieved from https://robomongo.org/
Thornton, M. O. (2011, August 19). *Bootstrap*.
Retrieved from https://getbootstrap.com/
Torvalds, L. (2014, December 17). *Git*.
Retrieved from https://git-scm.com/
VandeHei, J. (2017). *Axios*.
Retrieved from https://www.npmjs.com/package/axios

2. DESCRIPTION

2.1 PRODUCT PERSPECTIVE

- ✓ **User** : When the user logged in :
  - Can add comments, Ask Questions, Edit Profile Info.
  - Submit an application to Join Us.
- ✓ **Guest**: Can only view Events and the Technical, HR Questions without the answers.
- ✓ **HR/Technical Admin** :
  - Add New Post and Comments
  - Delete Post
  - Edit Profile Info.
  - Answer Questions.
- ✓ **Owner** : She/he can :
  - Add New Event, New User and Comments
  - Delete Users
  - Edit Profile Info.
  - Accept Users.

2.2 FEATURES

- ✓ Login: This function allows a registered user to log in to his account using his frequent email with the password. If a user is not registered, the website shall allow the user to enroll first. The system will check both the frequent email and password when a user attempts to log in.
- ✓ Logout: The logout section provides a way for the user to securely log out of the system. This process will save all user operations when he/she exits the system. If a user wishes to continue accessing the website, he must log-in again to access user features.
- ✓ User Profile: This section gives the user the power to view, edit the information stored in the user account. The user can check his quizzes marks, look at the status of asked question or JoinUs application that were requested. This feature is not available for non-registered users.
- ✓ Add new event: This section gives the user with role Owner the power to add an event on the website by submitting the event information.
- ✓ Add new post: This section gives the user with role Owner, admin the power to add new post on the website by submitting the post information.
- ✓ Admin Dashboard: This section gives the owner the power to view, delete and add users and admins, Add and delete event and accept pending applications.
- ✓ Join Us: This section gives the user the power to submit an application to join us as a new HR or Technical admin.
- ✓ Ask Question: This section gives the user the power to ask any question on any field. User has to have an account or to register before.
- ✓ About us: This section gives the user the power to view our Developer team and their LinkedIn, Facebook and GitHub account.
- ✓ HR Questions: This section gives the user the power to view all HR questions and answers and add comments on posts.
- ✓ Technical Questions: This section gives the user the power to view all Technical questions and answers and add comments on posts.
- ✓ User Dashboard: This section gives the user the power to view all his/her comments, asked question, pending questions and application and Delete comments.

2.3 USER OVERVIEW

The main actors in the system are Guest, User, Admin and Owner. Brief descriptions of these classes follow:

- ✓ User - The User is person has properties like Name, email, password, phone number Associated with field. Can log into the website and view any post depend on the fields such as HR or Technical , view events ,edit his profile, JoinUs, AskQuestion and add comments to any post.
- ✓ Admin -This person can add new question, show his pending questions, edit profile information ,delete questions depends on his role such as Technical or HR admin .
- ✓ Owner - This is the owner of head start. This person can add events, approve joining application requests, delete and add users, delete and add admins, edit profile information and delete any questions.

2.4 OPERATING ENVIRONMENT

- ✓ This website is designed to run any system that is capable of running ReactJS, NodeJS, Express and MongoDB.
- ✓ This covers a wide variety of machines, including operating systems Mac OS X, Windows XP, and Vista and Linux.
- ✓ The sole requirement for the user is a web browser (Safari, Google Chrome, Firefox or Internet Explorer) with an active internet connection.

2.5 CONTRAINTS: IMPLEMENTATION / DESIGN

- ✓ Minimum 4 GB RAM.
- ✓ 64-bit architecture.

2.6 DOCUMENTATION

- ✓ GitHub Repository.
- ✓ Google Drive Folder.

2.7 ASSUMPTIONS / DEPENDENCIES

- ✓ React Library.
- ✓ Cascading Style Sheets (CSS).
- ✓ Bootstrap Library.
- ✓ Express Framework.
- ✓ MongoDB document database.
- ✓ NodeJS Environment.
- ✓ Visual Studio Code Software.
- ✓ Postman Software.
- ✓ Git version control.
- ✓ GitHub hosting website.
- ✓ JavaScript.

3. SYSTEM FEATURES

3.1 Login Page Description:

- Providing the user with a page to login to the website.
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on Login Link.
- Response: Login Page is displayed
- Stimulus: User Enters Username and Password
- Response: Username and Password are validated from the Database.
- Stimulus: User Clicks on the Login Button
- Response: Landing Page is displayed if Username and Password are correct else Error Message is displayed.
- Functional Requirements:

- ✓ The user shall be able to view and click on the Login Link.
- ✓ The user shall be able to enter the username and password
- ✓ The database shall be able to validate a username and password.

3.2 View Questions Description:

- Providing the user with a page to view the Questions and answer of it.
- Priority = 9.
- Stimulus/Response Sequences:
- Stimulus: User clicks on HR/Technical Page Link
- Response: Questions are displayed. Read more button is displayed
- Stimulus: User clicks on a particular Question.
- Response: The question page and read more button.
- Functional Requirements.
    - ✓ The user shall be able to view and click on the HR/Technical Questions Link.
    - ✓ The user shall be able to view the details for each question.

3.3 View Profile Description:

- Provide the user with a page to view his information
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on Profile link
- Response: The profile is displayed.
- Stimulus: User Edit any input.

- Response: The edit button of his information is displayed.
- Stimulus: User clicks on edit.
- Response: The information will be edited.
- Functional Requirements
  - ✓ The user shall be able to view his profile and edit his information.

3.4 View About us

- The User shall be able to view the owner information.
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on about us page
- Response: The about us is displayed.
- Functional Requirements
  - ✓ The system shall be able to view about us.

3.5 Logout

- The user shall be able to Logout.
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on the Logout link
- Response: User is logged out and the index page is displayed.
- Functional Requirements
  - ✓ The user shall be able to logout from the System

3.6 Ask Questions

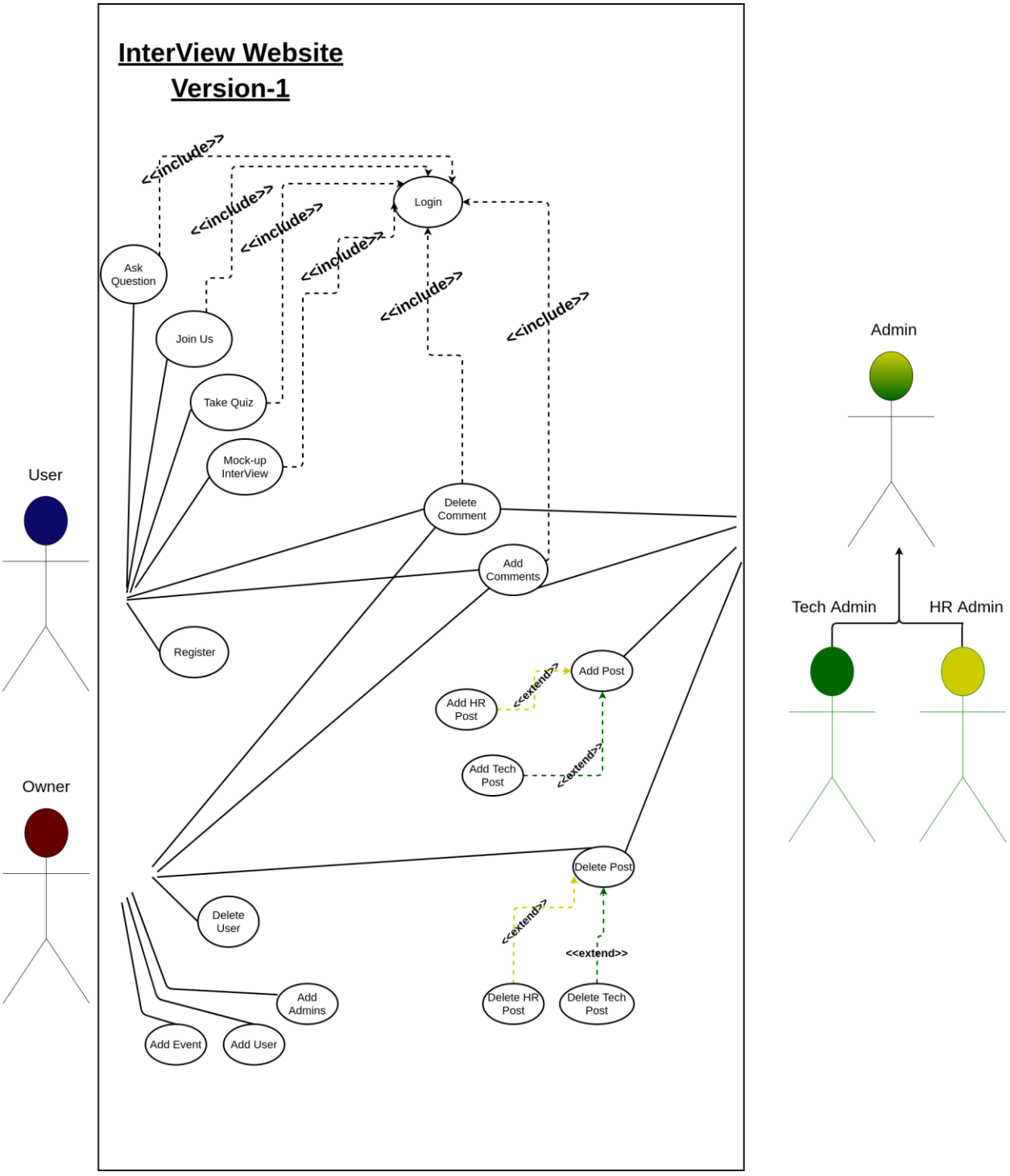
- The user shall be able to ask question.
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on the ask question link
- Response: The ask question page is displayed.
- Stimulus: User fills the form and click ask button.
- Response: Question will be sending to be answered.
- Functional Requirements
    - ✓ The user shall be able to ask question.

3.7 Join us

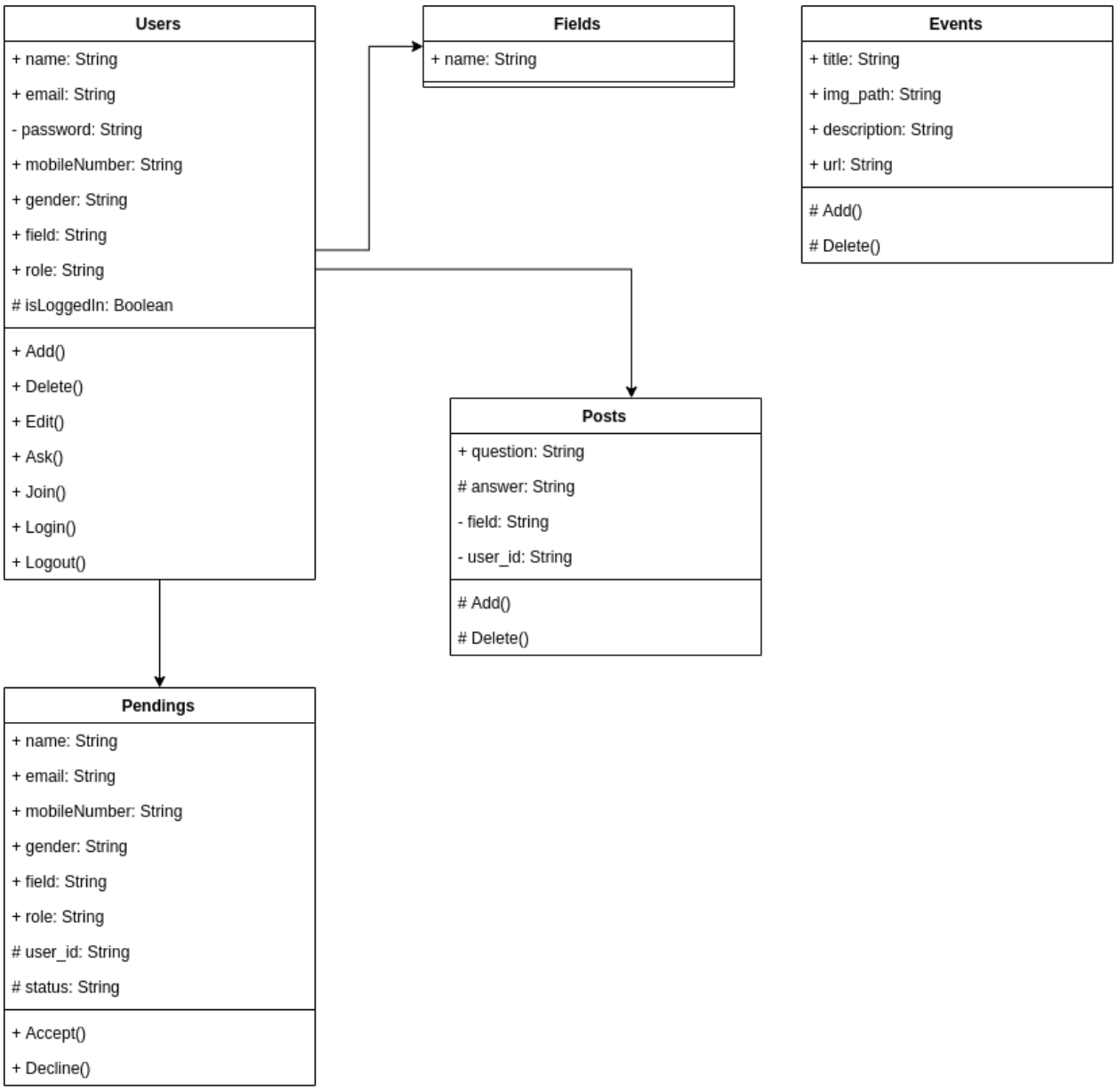
- The user shall be able to join us.
- Priority = 9
- Stimulus/Response Sequences:
- Stimulus: User clicks on the join us link.
- Response: The ask join us page is displayed.
- Stimulus: User clicks on submit application button.
- Response: Application will be sending to be reviewed .
- Functional Requirements
    - ✓ The user shall be able to submit an applications.

4. REQUIREMENTS OF EXTERNAL INTERFACE

4.1 Use Case Diagram:



4.2 Database Design:



4.3 USER INTERFACES

- ✓ The system shall provide all functionalities and activities supported
- ✓ The pages shall permit complete navigation and item selection.
- ✓ Provide User interface to view posts.
- ✓ Provide User interface to add comments.
- ✓ Provide User interface to edit information.

4.4 Design the database

4.4.1 SOFTWARE INTERFACES

Front-End-Side:

- ✓ Axios : 0.19.0
- ✓ Bootstrap : 4.3.1
- ✓ Cors : 2.8.5
- ✓ Firebase : 7.2.2
- ✓ JQuery : 3.4.1
- ✓ React : 16.10.2
- ✓ React-bootstrap : 1.0.0-beta.14
- ✓ React-dom : 16.10.2
- ✓ React-firebase : 2.2.8
- ✓ React-router-dom : 5.1.2
- ✓ React-scripts : 3.3.0

Back-End-Side:

- ✓ Cors: 2.8.5
- ✓ express: 4.17.1
- ✓ mongoose: 5.7.6
- ✓ Node: 12.12.0
- ✓ Nodemon: 1.19.4

4.4.2 COMMUNICATION INTERFACES

The system must utilize the standard Hypertext Transfer Protocol (HTTP) to ensure maximum inter-browser compatibility. The client accesses the system through a web browser.

5. ADDITIONAL NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE

The Head Start Website shall have capabilities to accept 500 connections. For each session, the system shall guarantee the connection time 5 minutes from the last input, after which the connection will be deemed expired. A close operation will be performed when expired. This design is to satisfy each user's usability and connection quality.

5.2 SAFETY

Checking the fact that all clients must be attached to one server, so there is an appropriate control of the information.

5.3 SECURITY

Passwords must be a minimum of eight digits and must contain one upper-case letter one lower-case letter. Phone number should be minimum 10 digits. This email must be a uniform email like text@text.com. All exchanges from client to server involving private data shall occur using the highest available level of secure connection (e.g., https)

5.4 SOFTWARE QUALITY

- ✓ The system shall be available to users all the time.
- ✓ The system shall always have something to function and always pop up error messages in case of component failure.
- ✓ The system shall always pop up message after executing any function.
- ✓ The system shall generate the correct pages with an accuracy of 99%.
- ✓ The system shall provide the right tools to support all its features.

6. APPENDICES

6.1 APPENDIX A: Wireframe:

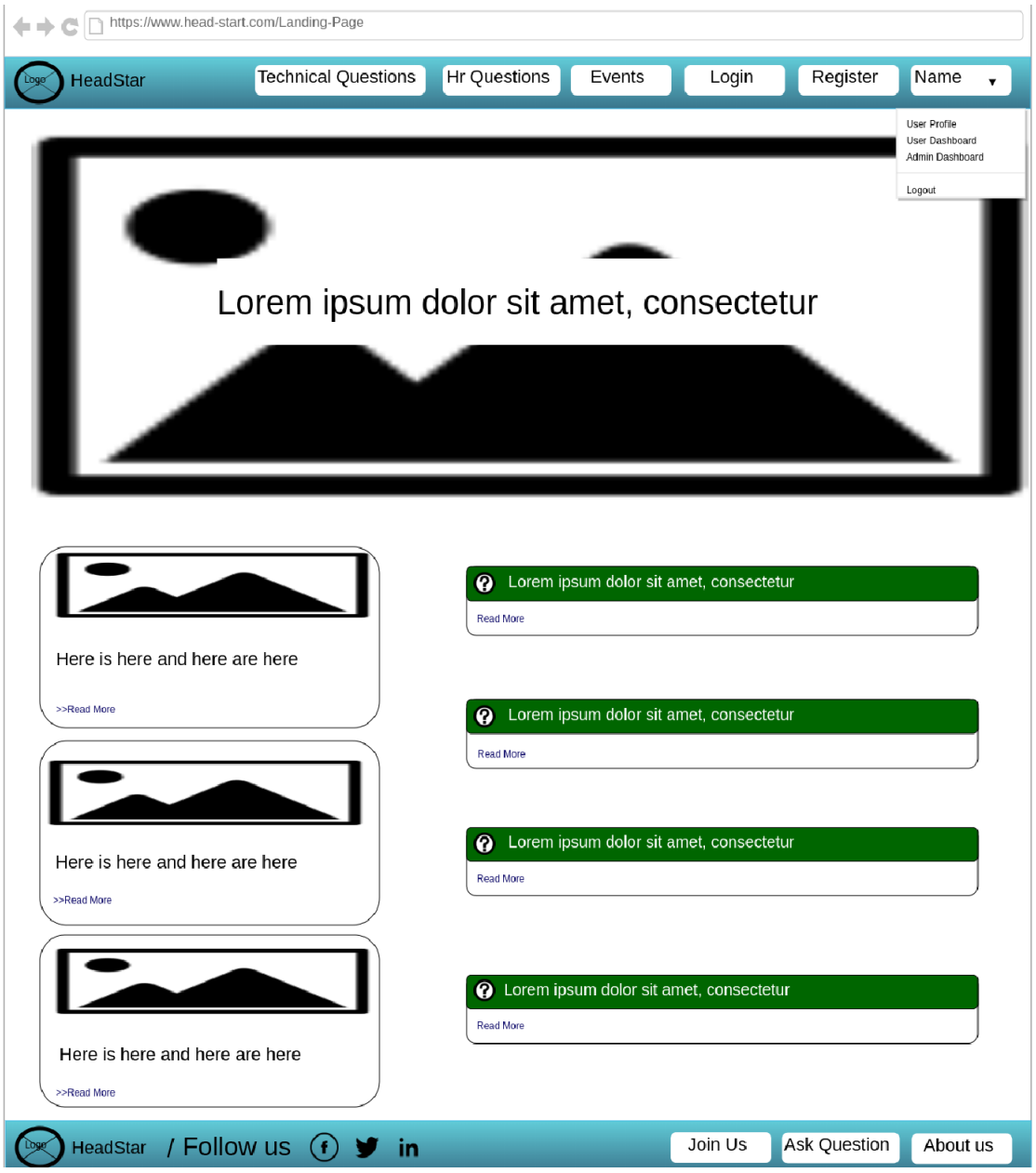


Figure 1 :Landing Page

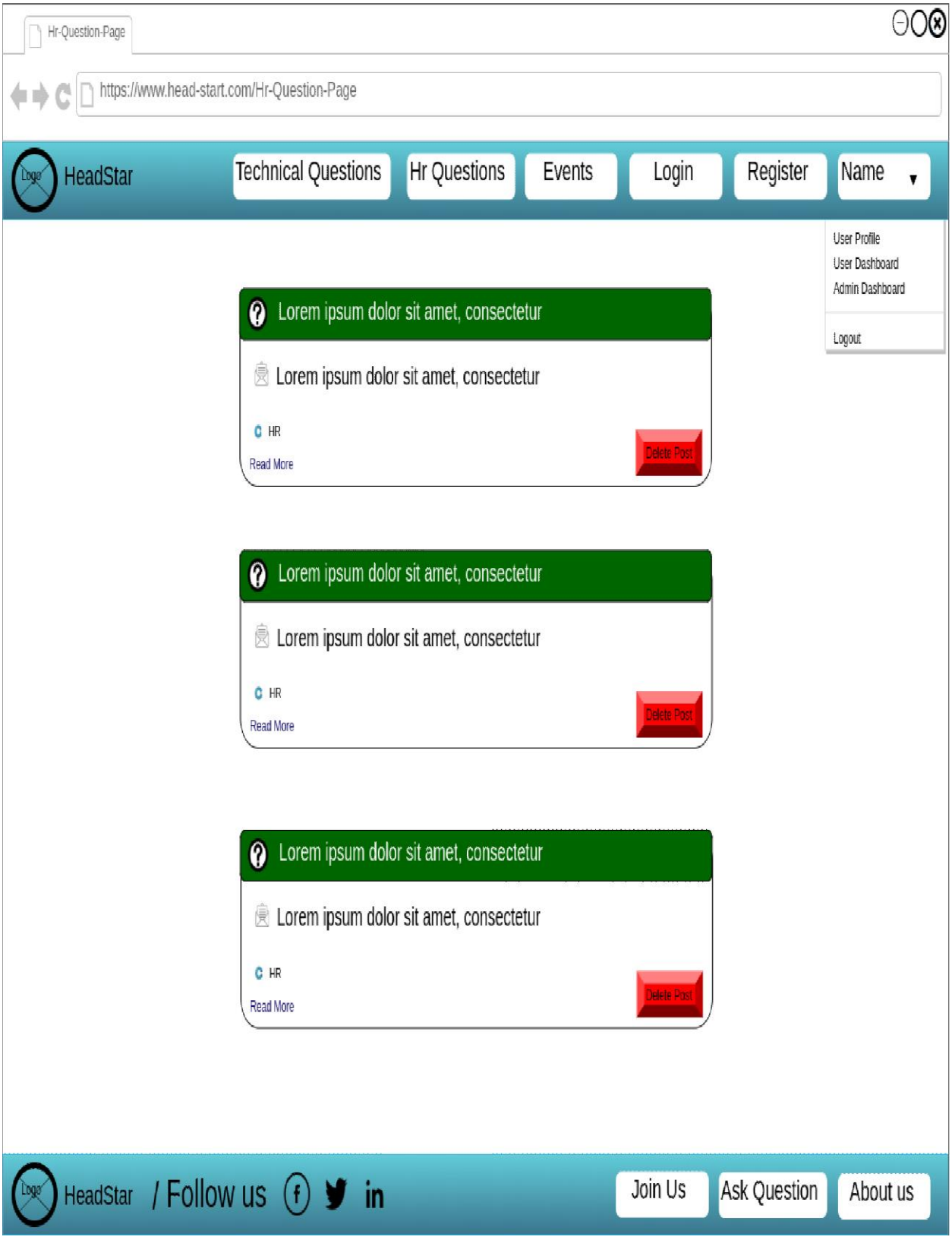


Figure 2 :HR Question Page

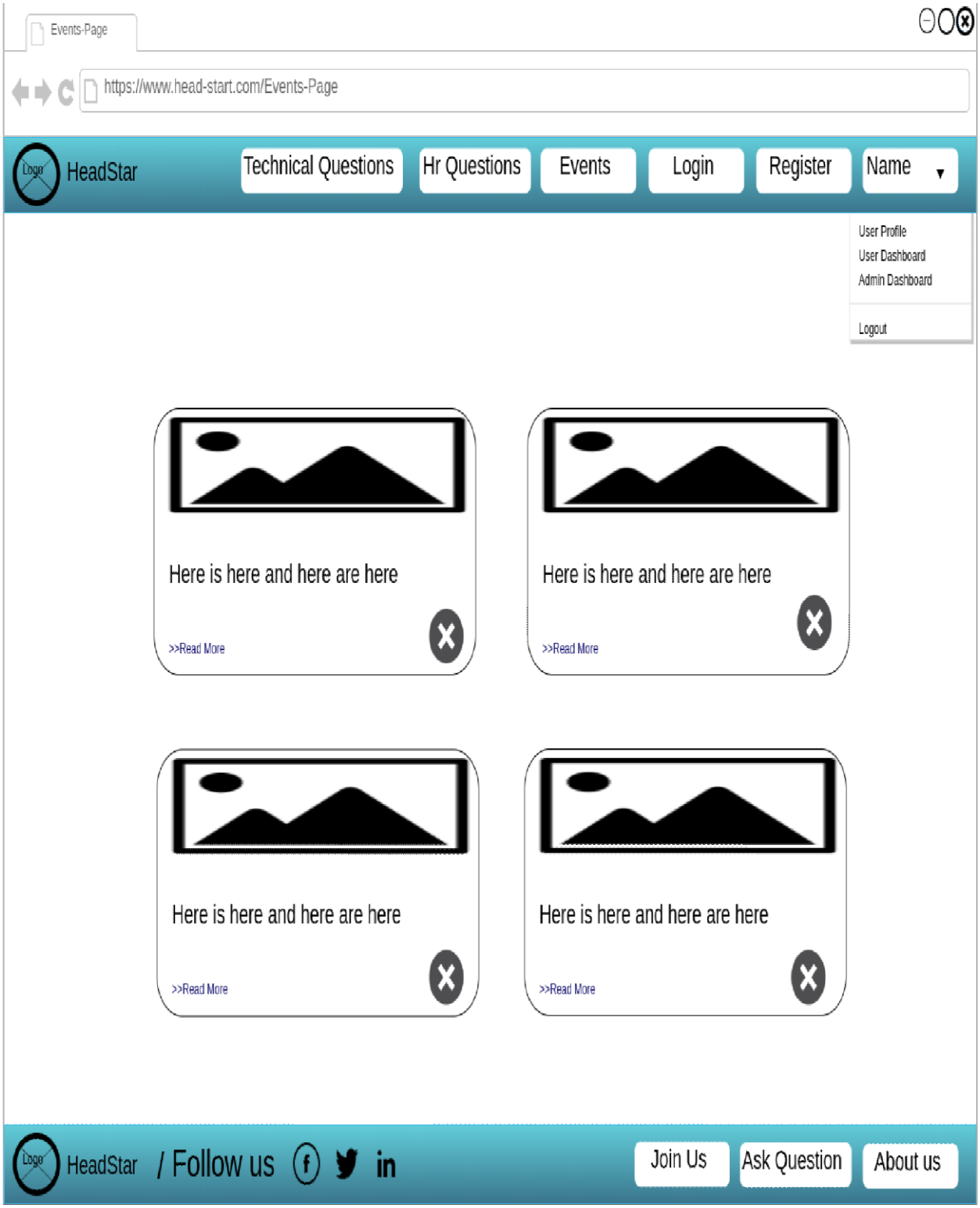


Figure 3: Events Page

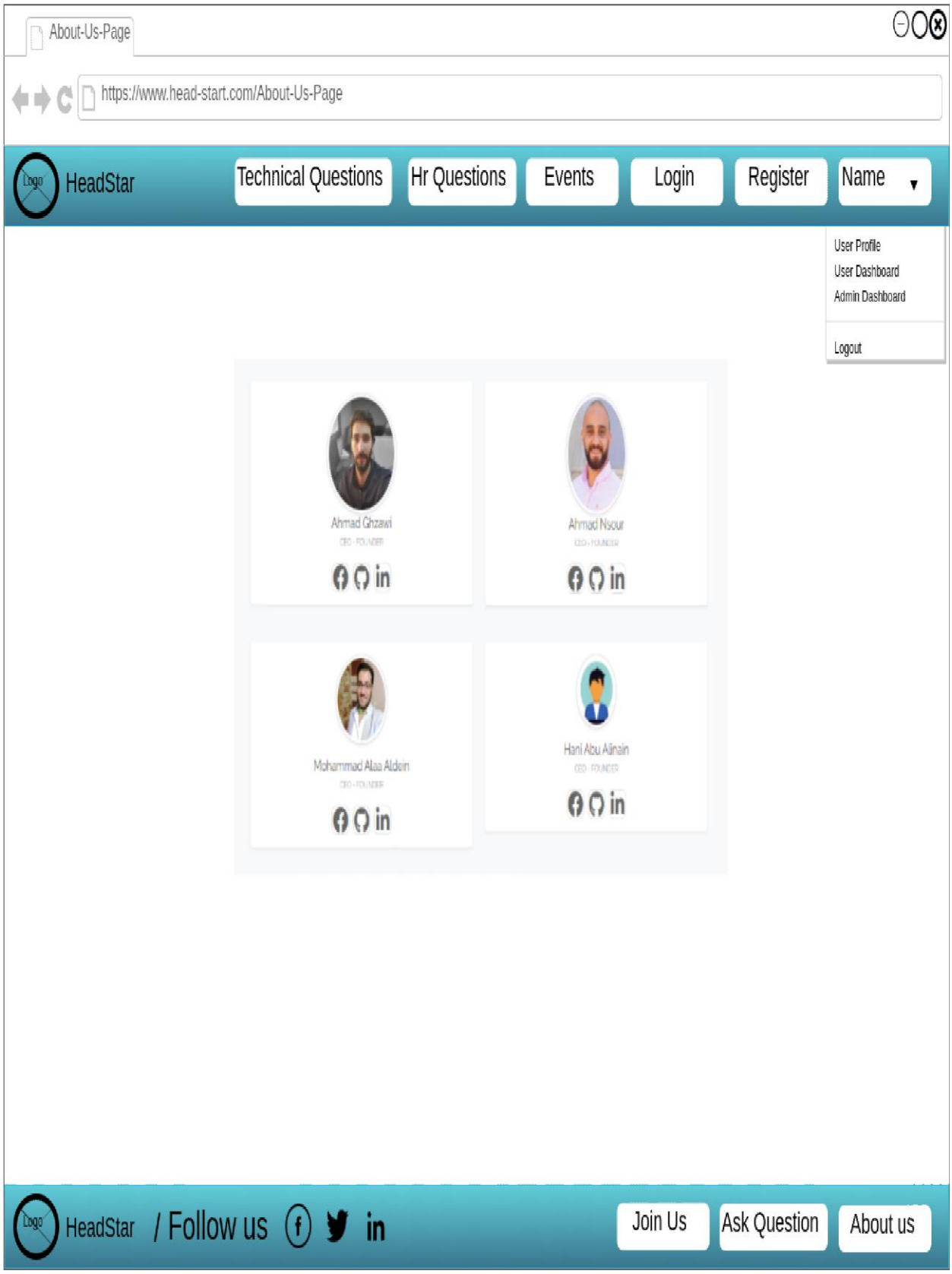


Figure 4: About Us Page

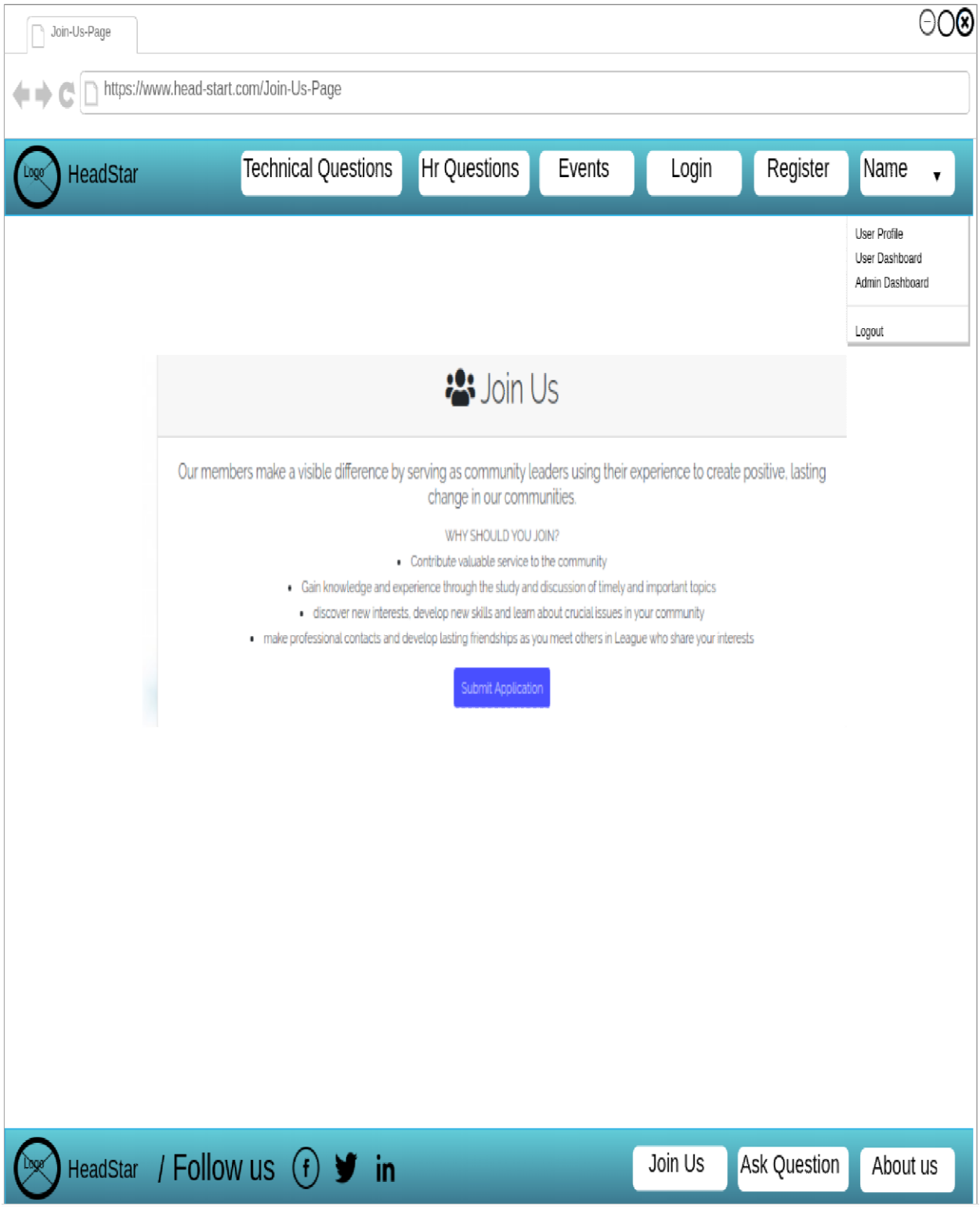


Figure 5 :Submit An Application

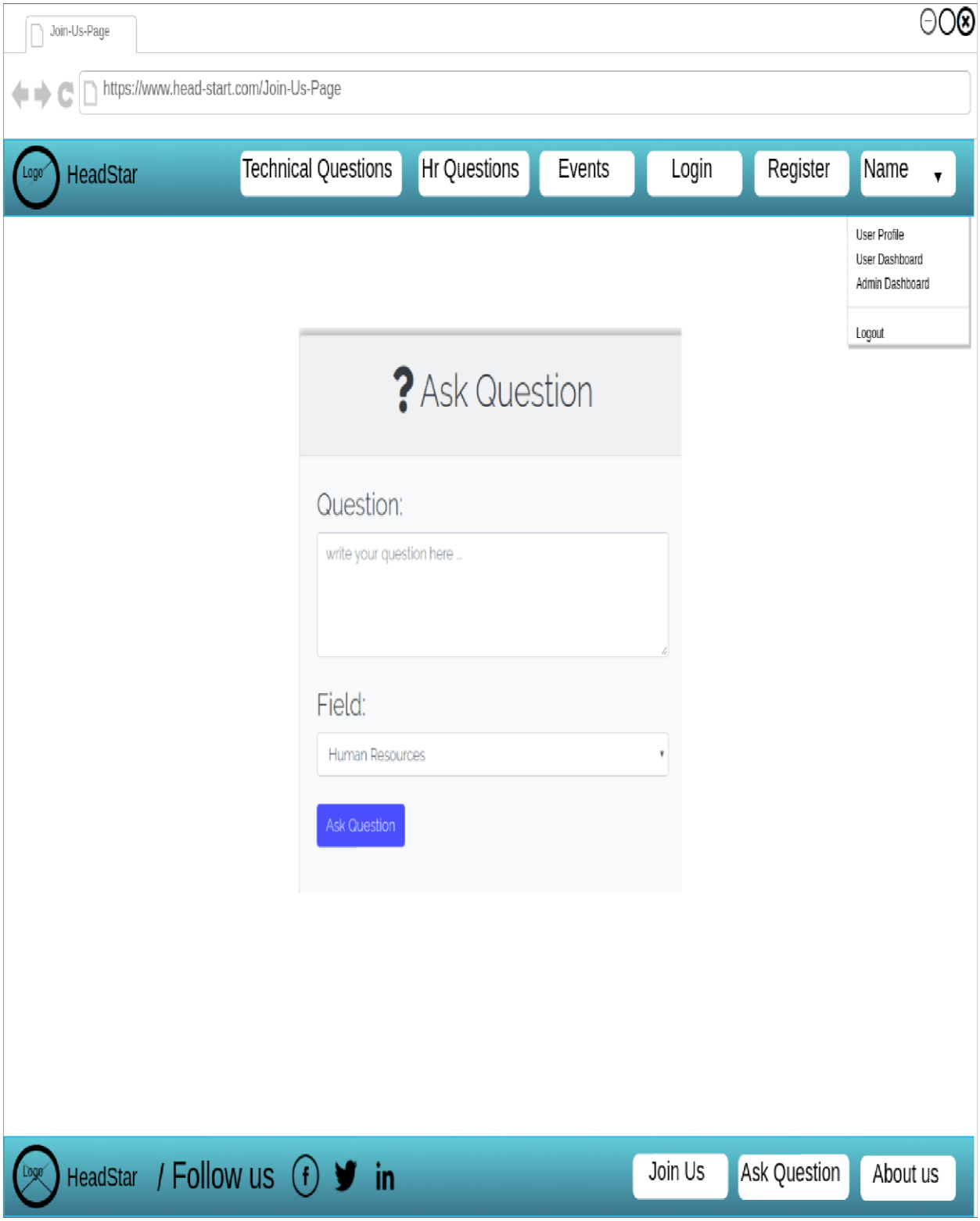


Figure 6: Ask Question Page

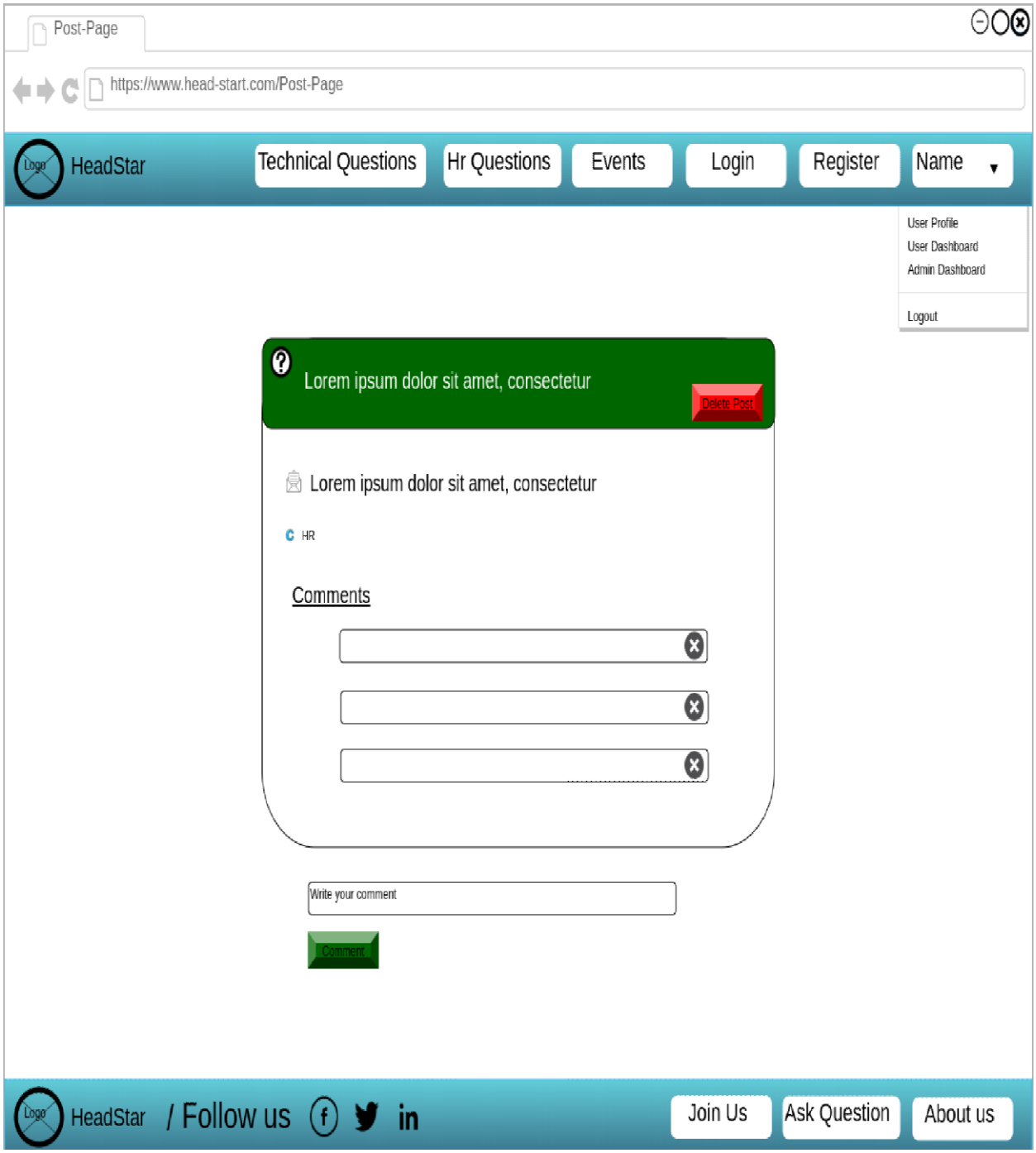


Figure 7: Post Page

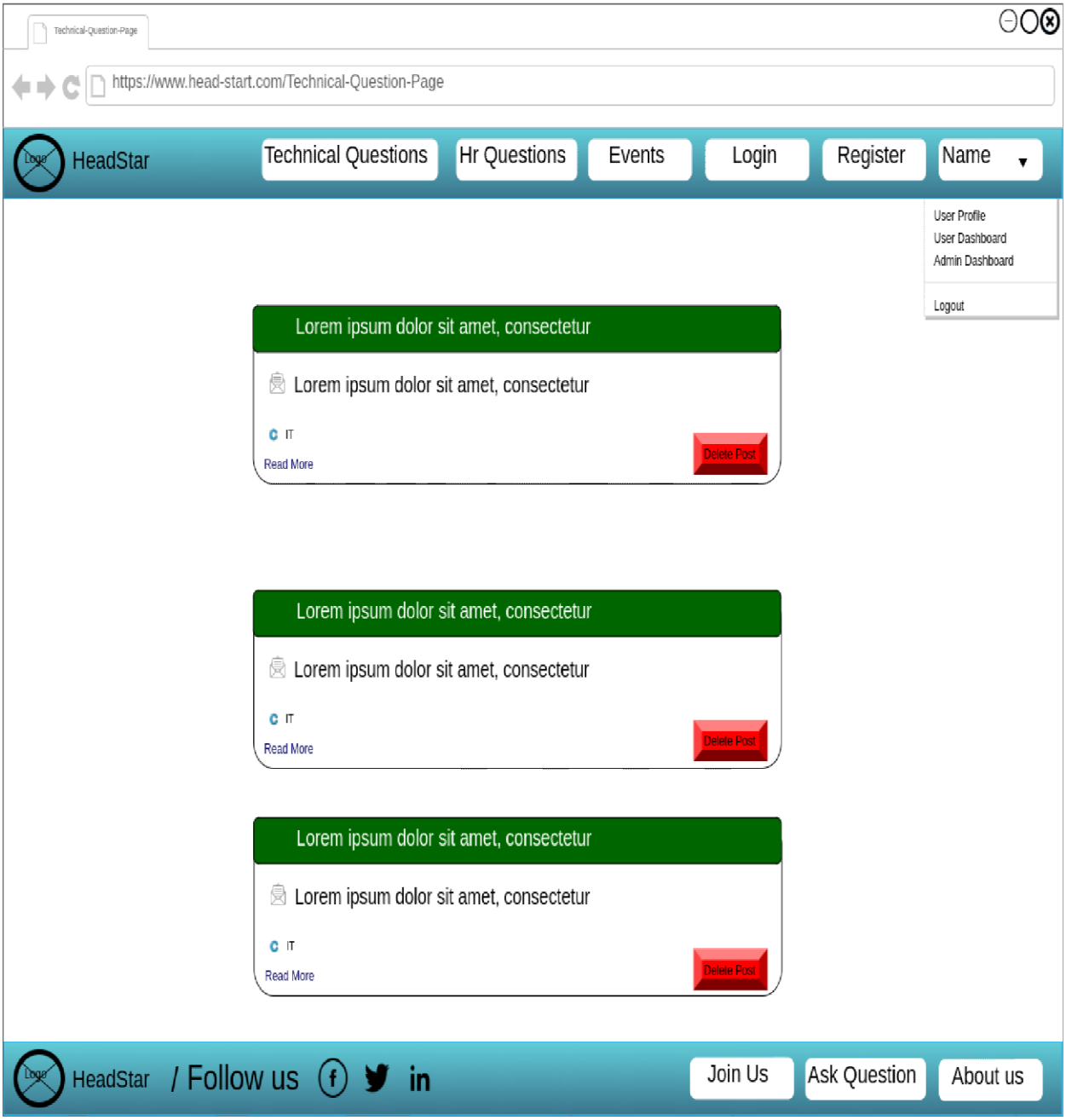


Figure 8 :Technical Question Page

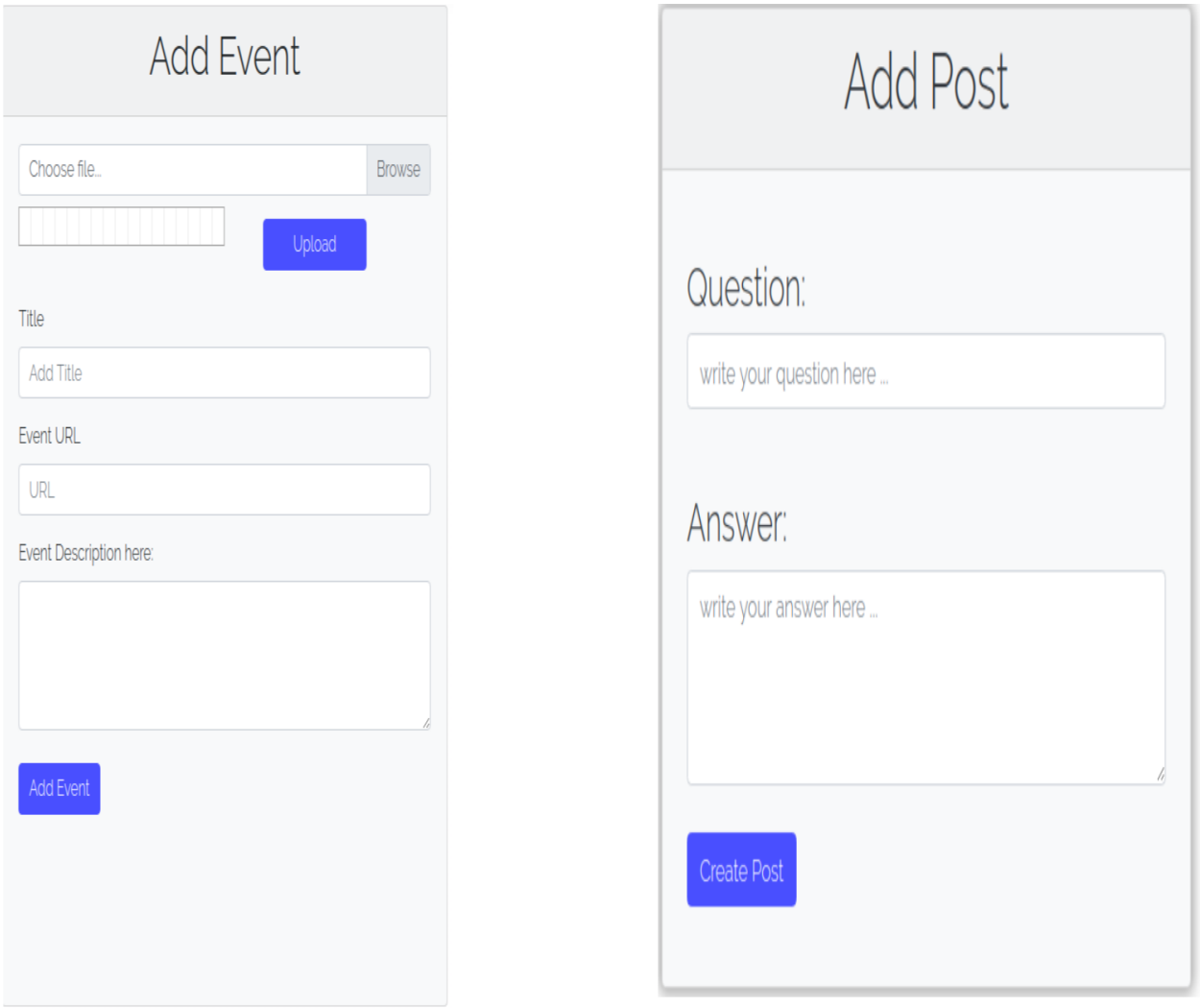


Figure 9: Add Event and Add Post Pages

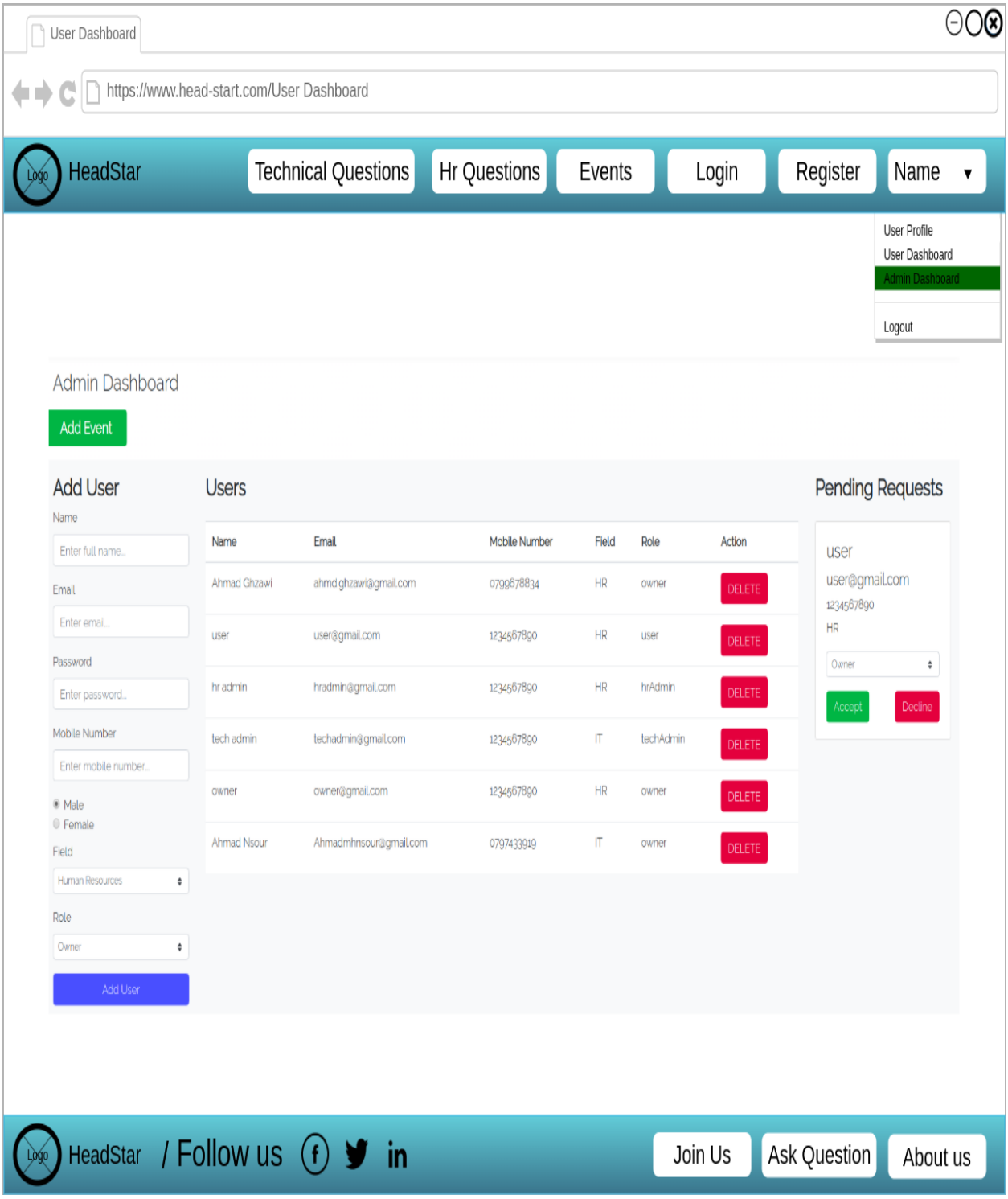


Figure 10: Admin Dashboard