System Requirements Specification Index

For

Interview Tracker Application (InMemory)

Version 4.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
fullstack@iiht.com

INTERVIEW TRACKER APPLICATION SYSTEM

System Requirements Specification

1. Business-Requirement:

1.1 Problem Statement:

The purpose of this application is to allow the official staff to register, Update, delete, show all interviewees by using **user API**. On the other hand, Dashboard **API** allows you to view all Interviews, search Interview by name or interviewer name, Add, Delete, Update interview and Using **Interview API** you can Add New Interview and count total number of interviews.

1.2 Following is the requirement specification:

	Interview Tracker Application
User Controller	1. Register new User/Interviewee/Interviewer.
	2. Allows to edit/update/Delete User/Interviewee/Interviewer profile.
Interview Controller	1. Show Total number of interviews.
	2. Add a new interview.
Dashboard Controller	1. Show All Interview on page for admin/Interviewer.
	2. Add a new interview.
	3. Update/Edit interview.
	4. Remove interview.
	5. Get the total count of the interview.
	6. Find an interview by Interview/Interviewer Name.

2.Resources available:

2.1 Package: InterviewTracker

Names	Resource	Remarks	Status
Package Structure			
controller	User, Dashboard, Interview Controller	These controllers handle all application Function, update/Edit show information and login existing user.	Partially Implemented
Startup.cs	Startup CS file	Contain all Services settings and Db Configuration.	Already Implemented
Properties	launchSettings.json file	All URL Setting for API	Already Implemented

2.2 Package: InterviewTracker.BusinessLayer

Names	Resource	Remarks	Status
Package Structure			
Interface	IInterviewTracker, IUserInterviewTracker Services interface	Inside all these cs files contains all business logic functions.	Already Implemented
Service	InterviewTracker, UserInterviewTracker Servicesclass file	Using this all class we are calling the Repository method and use it in the program and on the controller.	Partially Implemented
Repository	IInterviewTracker, InterviewTracker, IUserInterviewTracker, UserInterviewTracker Repository CS file and interface.	All these interfaces and class files contain all CRUD operation code for Db.	Partially Implemented
ViewModels	AddInterview, EditInterview, Interview, Register, UserEdit ViewModel Class file	Contain all view Domain entities for show and bind data.	Already Implemented

2.3 Package: InterviewTracker.dataLayer

Names	Resource	Remarks	Status
Package Structure			
DataLayer	Db Settings, InterviewTrackerDbConte xt	All database setting class	Already Implemented

2.4 Package: InterviewTracker.Entities

Names	Resource	Remarks	Status
Package Structure			
Entities and Enum	ApplicationUser, Interview, InterviewStatus, Status, TechnicalInterviewStatus, UserType CS and Enum file	All Entities/Domain attribute	Already Implemented

2.5 Package: InterviewTracker.Tests

Note: - Under the InterviewTrackerTests.Tests project contains all test case classes and functions for code evaluation. Don't edit or change anything inside this project.

3. Rest Endpoints

Rest End-points to be exposed in the controller along with method details for the same to be created

3.1 UserController

URL Exposed		Purpose
/user		Fetch all User
Http Method	GET	
Parameters	-	
Return	<IEnumerable<ApplicationUser>>	
/user/RegisterUser		Register new User
Http Method	POST	
Parameter 1	ApplicationUser user	
Return	HTTP Response StatusCode	
/user/Updateuser/{UserId}		Update/Edit User based on id
Http Method	PUT	
Parameter 1	Int(userId)	
Return	HTTP Response StatusCode	
/user/DeleteUser/{UserId}		Delete User based on id
Http Method	DELETE	
Parameter 1	Int (UserId)	
Return	HTTP Response StatusCode	

3.2 InterviewController

URL Exposed		Purpose
/interview		Fetch total number of interview
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Interview>	
/interview/AddInterview		Add new Interview
Http Method	POST	
Parameter 1	AddInterviewViewModel	
Return	HTTP Response StatusCode	

3.3 DashboardController

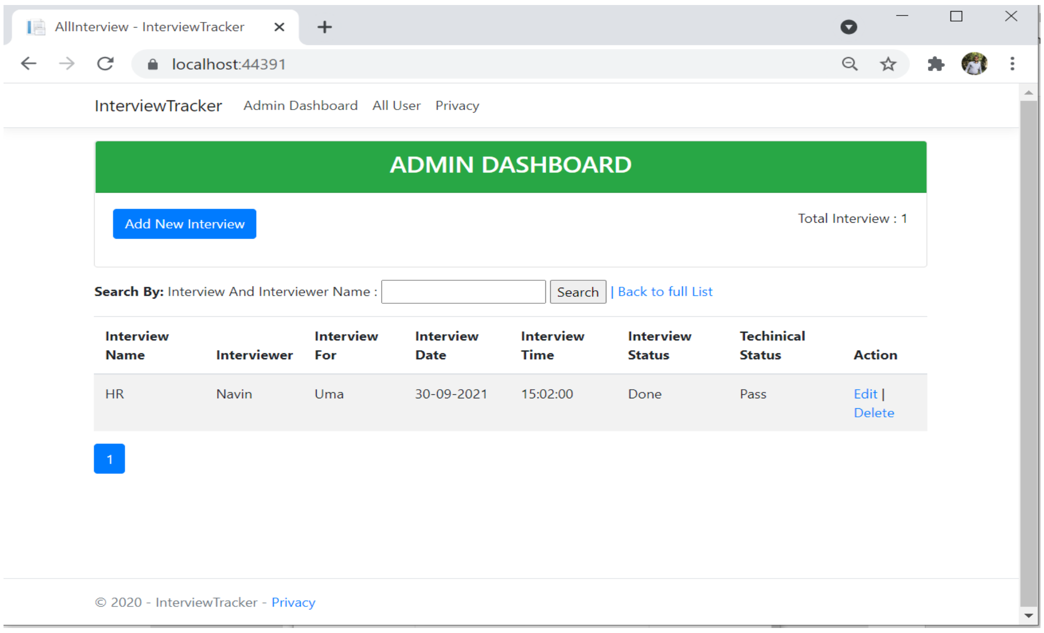
URL Exposed		Purpose
/dashboard		Fetch all Interview
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Interview>>	
/interview/DeleteInterview/{InterviewId}		Delete interview
Http Method	DELETE	
Parameter 1	Int(InterviewId)	
Return	HTTP Response StatusCode	
/interview/UpdateInterview/{InterviewId}		Update Existing Interview
Http Method	POST	
Parameter 1	Int(InterviewId)	
Return	HTTP Response StatusCode	
/interview/GetInterview/{InterviewId}		Fetch single interview by Id
Http Method	GET	
Parameter 1	Int(InterviewId)	
Return	<Interview>	

/interview/SearchInterview/{Name} Http Method | GET Parameter 1 | String(name) Return | <IEnumerable<Interviewt> >	Find interview by interviewer and Interview name
/interview/TotalInterview Http Method | GET Parameter 1 | Return |	Count total interview.

4. wireframe:

1. Following wireframe need to consider to develop the UI.

Home Page.



ADMIN DASHBOARD

Add New Interview

Total Interview : 1

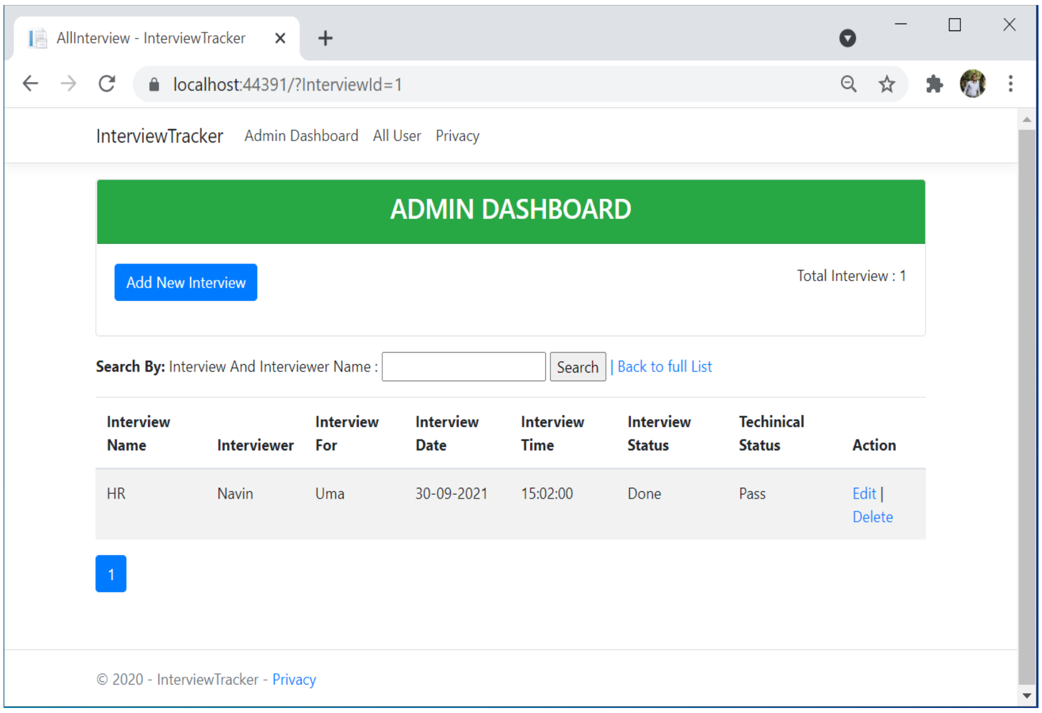
Search By: Interview And Interviewer Name : Search | Back to full List

Interview Name	Interviewer	Interview For	Interview Date	Interview Time	Interview Status	Techinical Status	Action
HR	Navin	Uma	30-09-2021	15:02:00	Done	Pass	Edit | Delete

1

© 2020 - InterviewTracker - Privacy

Interview By Id Page



AllInterview - InterviewTracker

localhost:44391/?InterviewId=1

InterviewTracker Admin Dashboard All User Privacy

ADMIN DASHBOARD

Add New Interview

Total Interview : 1

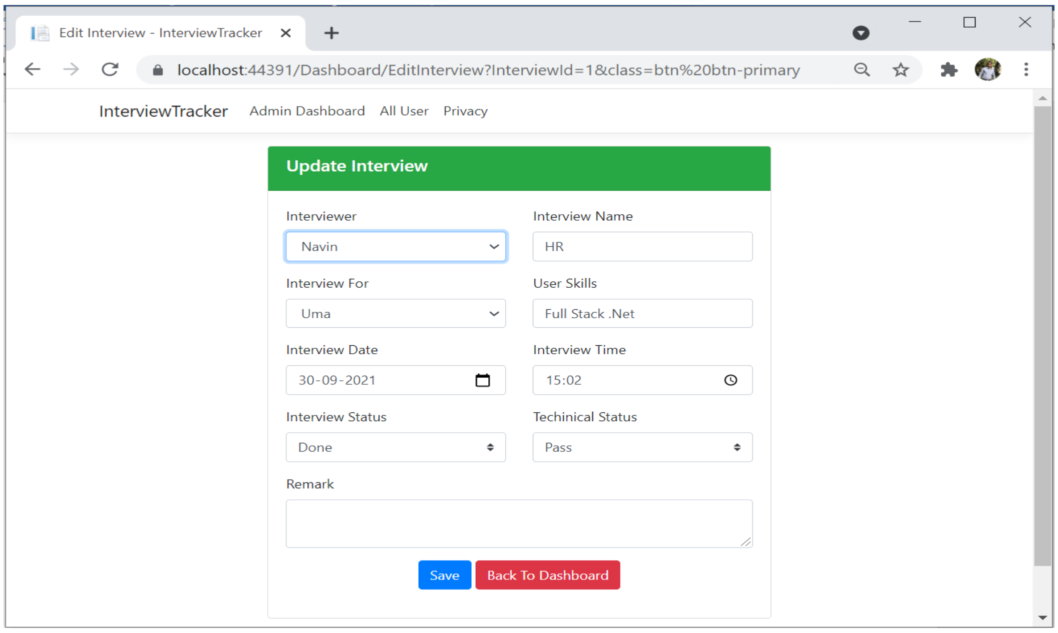
Search By: Interview And Interviewer Name : Search | Back to full List

Interview Name	Interviewer	Interview For	Interview Date	Interview Time	Interview Status	Techinical Status	Action
HR	Navin	Uma	30-09-2021	15:02:00	Done	Pass	Edit | Delete

1

© 2020 - InterviewTracker - Privacy

Edit/Update Interview page



Edit Interview - InterviewTracker

localhost:44391/Dashboard/EditInterview?InterviewId=1&class=btn%20btn-primary

InterviewTracker Admin Dashboard All User Privacy

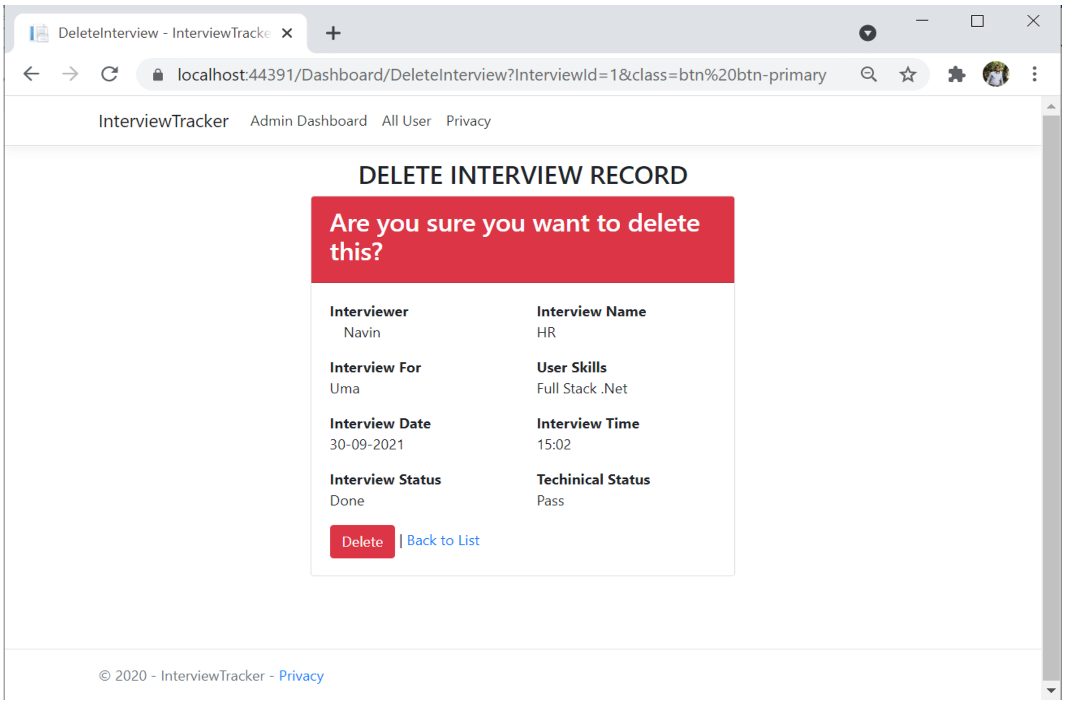
Update Interview

Interviewer: Navin	Interview Name: HR
Interview For: Uma	User Skills: Full Stack .Net
Interview Date: 30-09-2021	Interview Time: 15:02
Interview Status: Done	Techinical Status: Pass

Remark

Save Back To Dashboard

DeleteInterview Page



DeleteInterview - InterviewTracker

localhost:44391/Dashboard/DeleteInterview?InterviewId=1&class=btn%20btn-primary

InterviewTracker Admin Dashboard All User Privacy

DELETE INTERVIEW RECORD

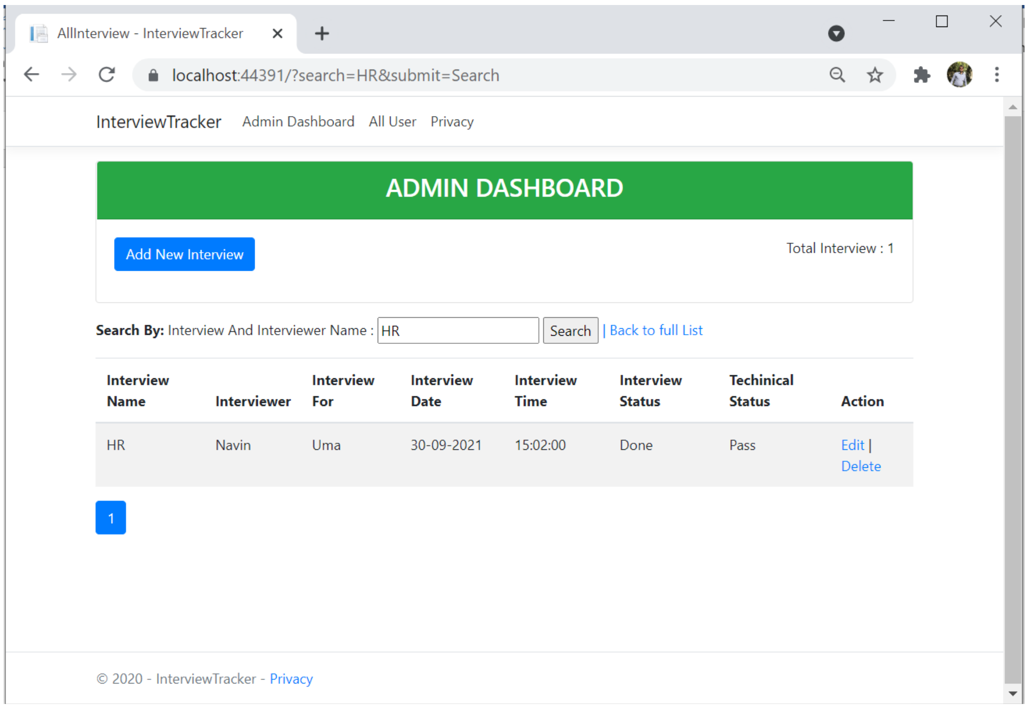
Are you sure you want to delete this?

Interviewer	**Interview Name**
Navin	HR
Interview For	**User Skills**
Uma	Full Stack .Net
Interview Date	**Interview Time**
30-09-2021	15:02
Interview Status	**Techinical Status**
Done	Pass

Delete | Back to List

© 2020 - InterviewTracker - Privacy

Search/Find Interview result Page



AllInterview - InterviewTracker

localhost:44391/?search=HR&submit=Search

InterviewTracker Admin Dashboard All User Privacy

ADMIN DASHBOARD

Add New Interview

Total Interview : 1

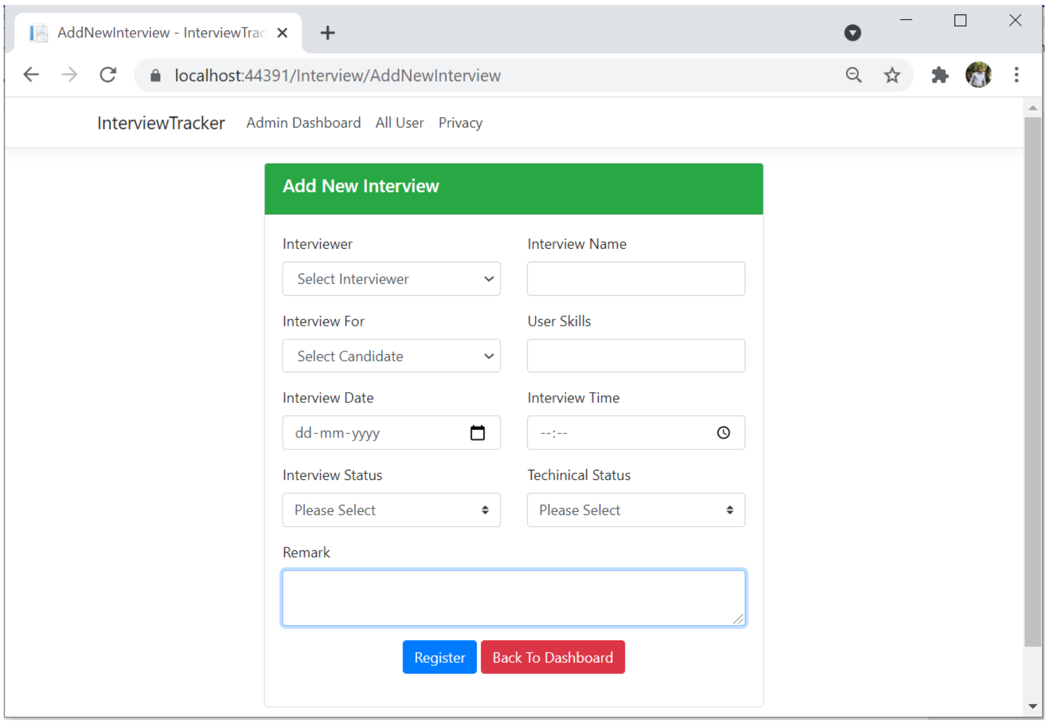
Search By: Interview And Interviewer Name : HR Search | Back to full List

Interview Name	Interviewer	Interview For	Interview Date	Interview Time	Interview Status	Techinical Status	Action
HR	Navin	Uma	30-09-2021	15:02:00	Done	Pass	Edit | Delete

1

© 2020 - InterviewTracker - Privacy

Addnew Interview Page

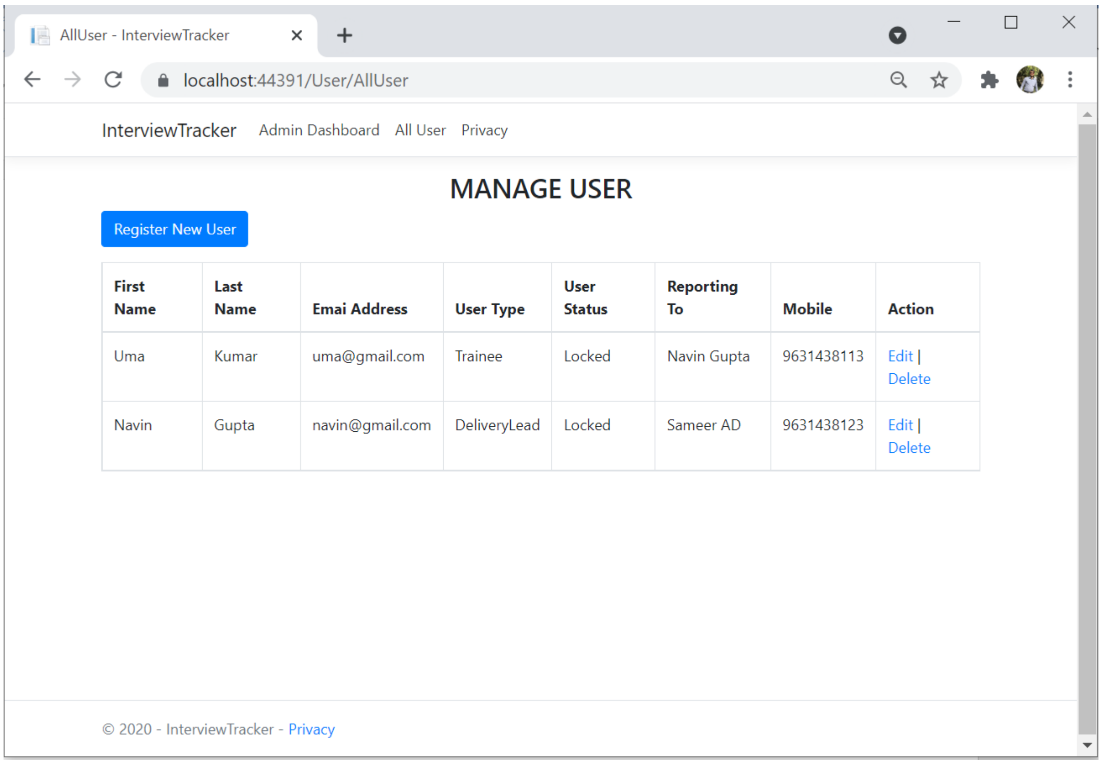

AddNewInterview - InterviewTrac
localhost:44391/Interview/AddNewInterview
InterviewTracker
Admin Dashboard
All User
Privacy

Add New Interview

Interviewer: Select Interviewer	Interview Name:
Interview For: Select Candidate	User Skills:
Interview Date: dd-mm-yyyy	Interview Time: --:--
Interview Status: Please Select	Techinical Status: Please Select
Remark:	

Register
Back To Dashboard

AllUser Page

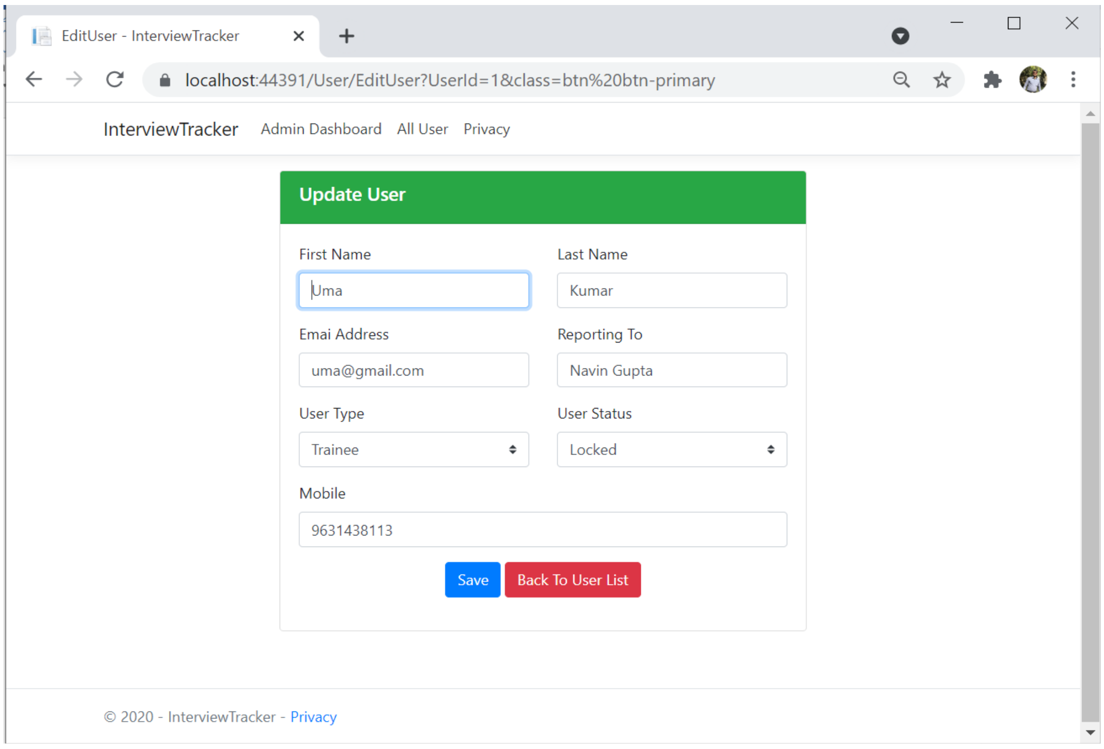


MANAGE USER

First Name	Last Name	Emai Address	User Type	User Status	Reporting To	Mobile	Action
Uma	Kumar	uma@gmail.com	Trainee	Locked	Navin Gupta	9631438113	Edit | Delete
Navin	Gupta	navin@gmail.com	DeliveryLead	Locked	Sameer AD	9631438123	Edit | Delete

© 2020 - InterviewTracker - Privacy

Edit/Update User Page

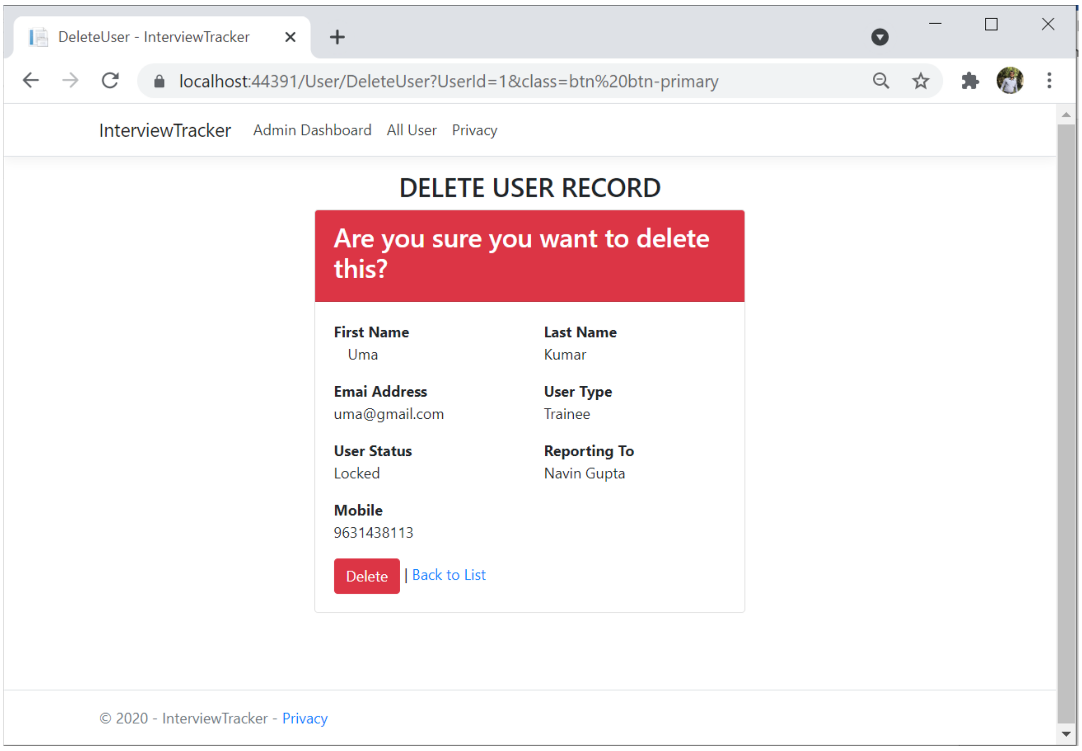

EditUser - InterviewTracker
localhost:44391/User/EditUser?UserId=1&class=btn%20btn-primary
InterviewTracker
Admin Dashboard
All User
Privacy

Update User

First Name	Last Name
Uma	Kumar
Emai Address	Reporting To
uma@gmail.com	Navin Gupta
User Type	User Status
Trainee	Locked
Mobile	
9631438113	

Save
Back To User List
© 2020 - InterviewTracker - Privacy

Delete User Page


DeleteUser - InterviewTracker
localhost:44391/User/DeleteUser?UserId=1&class=btn%20btn-primary
InterviewTracker
Admin Dashboard
All User
Privacy

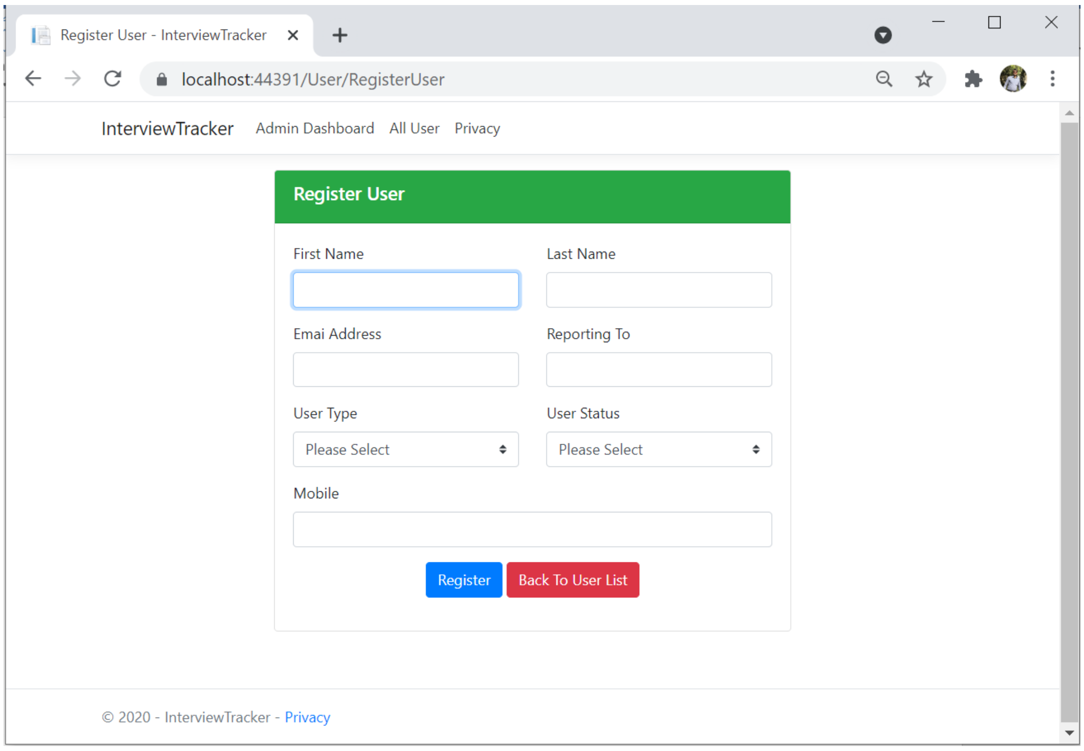
DELETE USER RECORD

Are you sure you want to delete this?

First Name	**Last Name**
Uma	Kumar
Emai Address	**User Type**
uma@gmail.com	Trainee
User Status	**Reporting To**
Locked	Navin Gupta
Mobile	
9631438113	

Delete
| Back to List
© 2020 - InterviewTracker - Privacy

Register New User Page



5. Business Validations

5.1 Common Constraints:

Following validation constraints are to be added

All the value for user details: must not be null.

All the Interview value: must not be null

Check Enum for all Fixed data under Entities Project

On Update or delete: must not be null of Id and Model

For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid

Must not go and touch the test resources, as they will be used for Auto-Evaluation.

Database used is Embedded InMemory.

6. Execution Steps to Follow

1. All actions like build, compile, running application, running test cases will be through Command Terminal.

2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) Terminal → New Terminal.

3. On command prompt, cd into your project folder (cd <Your-Project-folder>).

4. To build your project use command:
   (InterviewTrackerManagement /dotnet build)

5. To launch your application, Run the following command to run the application:
   (InterviewTrackerManagement / dotnet run)

6. This editor Auto Saves the code.

7. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.

8. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

9. To run the test cases in CMD, Run the following command to test the application:
   (InterviewTrackerManagement /dotnet test --logger "console;verbosity=detailed")
   (You can run this command multiple times to identify the test case status,
   and refactor code to make maximum test cases passed before final submission)

10. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B - command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.

11. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

12. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**
