Nuwan Athukorala - 20284580.docx

by Nuwan Theekshana Wickramarachi Athukorala

Submission date: 07-Sep-2023 07:24PM (UTC+0100)

Submission ID: 212359672

File name: 126754_Nuwan_Theekshana_Wickramarachi_Athukorala_Nuwan_Athukorala_-_20284580_2005213_353596474.docx (2.05M)

Word count: 1943

Character count: 11827

Task A

System Requirement Specification

Introduction

The Colombo Jobs Consulting Centre offers free consulting services to job searchers interested in working overseas. To keep records of employment consultants, job searchers, and appointments, the Centre currently depends on manual, paper-based procedures. Due to an increase in the number of job seekers requesting consultations, a contemporary, web-based system is required to simplify operations and provide better service. This System Requirement Specification (SRS) specifies the requirements for the proposed web-based system's development.

Purpose

The system's goal is to establish an effective and user-friendly platform for managing job consultants, job searchers, and appointments at the Jobs Consultation Centre. This system is intended to replace the current manual paper-based approach and will give the following advantages:

- Allow job searchers to book appointments online.
- Improve the process of keeping and retrieving records.
- Allow consultants to schedule their own time.
- Create reports that help managers make better decisions.

Objectives

The system's objectives are as follows:

- To make appointment scheduling easier for job searchers.
- To keep an up-to-date record of consultants' information and availability.
- To improve data security and access.
- The creation of reports enables efficient and informed decision-making.
- To provide an easy-to-use interface for both job seekers and consultants.

Functional Requirements

Admin

- User account management
- Access management
- View and update consultant, job seeker, and receptionist profiles
- View scheduled consultations
- Add country and job category details
- Report generation

Job Seeker

- Manage and update profile
- Search available consultants, countries and job categories
- Make consultation requests by selecting consultant, country, job category, date and time
- View consultation requests

Consultant

- Update consultant schedules
- View and manage consultation requests
- View job seeker profiles

Receptionist

- Manage job seeker and consultant profiles
- Make consultation requests on behalf of the job seeker
- View consultation requests
- Report generating

Non-functional requirements

Performance : Even during peak periods of usage, the website should load in less than three seconds. This ensures no long waits or delays when accessing the website.

Scalability: The website must be able to manage a large number of users and appointments without losing performance, security, or dependability. This ensures that the platform can manage increased demand and usage without becoming unstable.

Usability : The website should be easy to use and provide accurate information. This ensures that job seekers can easily navigate the site and request job consultations.

Compatibility: The website should be compatible with a wide range of browsers and devices so that users may access the platform regardless of their favourite operating system or technology.

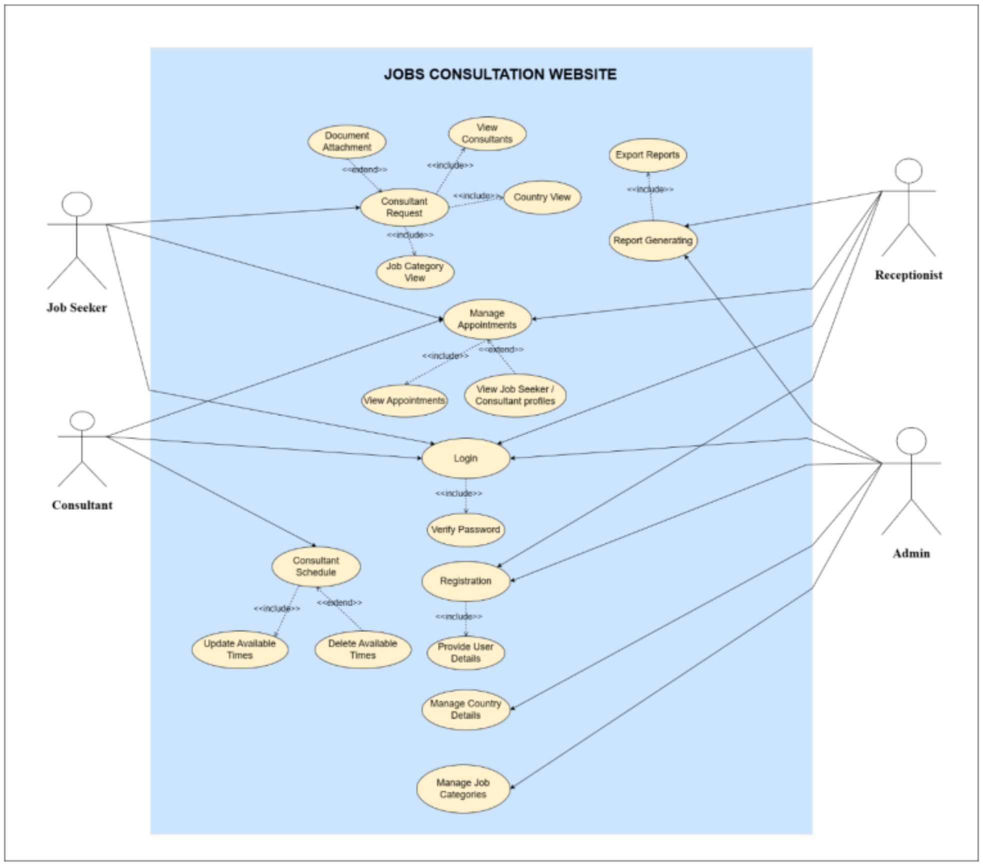
Security: The website should utilize authentication and encryption technology to secure user information and personal data. This safeguards users' privacy.

Task B

Use case diagram

A Use Case Diagram represents the many interactions or activities that a system or software program might have with external entities known as actors. It shows how users (actors) interact with the system through multiple use cases, which represent particular actions or scenarios, to offer a high-level understanding of the system's functionality. In the jobs consultation website mainly four actors are available as

- Admin
- Job seeker
- Consultant
- Receptionist



Use cases.

- **Login** – Users can login to the system with their email address and password
- **Registration** – Admin can create consultant and receptionist accounts. Receptionist can create consultant and job seeker accounts. Job seekers can also register in the website.
- **Consultant request** – Job seekers can make consultant requests after viewing country, job category and consultants.
- **Consultant schedule management** – Consultants can update available times and delete available times.
- **Manage appointments** – Receptionists, job seekers and consultants can view their appointments and view job seeker as well as consultant profiles.
- **Report generating** – Admin and receptionist can generate reports based on their needs

- **Managing country details** – Admin can add, update, delete, view consultation available countries.
- **Managing job categories** - Admin can add, update, delete, view consultation available job categories.
- **Managing user profiles** – All users can view and update user profiles and information.

Class diagram

Class Diagrams show the classes, objects, and their connections in a system or software program. These diagrams are used to represent the static structure of a system, including its classes, properties, methods, and class connections.

Sequence diagram[3]

A Sequence Diagram is a visual representation that shows how items and components in a system interact with one another in chronological order. It displays the flow of messages or actions through time and is used to simulate a system's dynamic behavior, notably the interactions between multiple components or players.

Job Seeker Sequence diagram

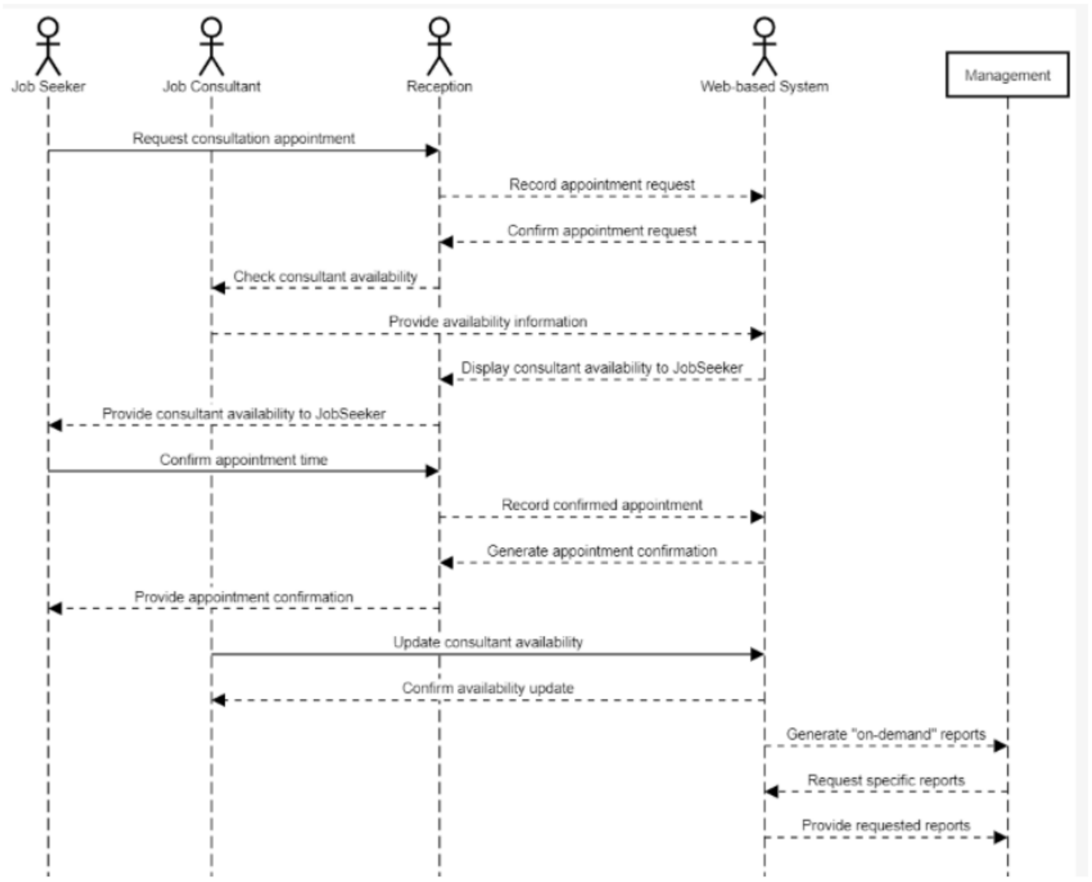
- Job sequence diagram for the job seeker can be broken down as follows
- The job seeker visit the website and register using his/her details

- Then the user is redirected to the login page and he/she enters the username and password to login
- The job seeker can then browse the website and filter the countries, job categories and consultants
- After filtering the job seeker can make a consultation request

Admin sequence diagram

- Admin can login to the system using the email and password
- Admin can choose different functions managing jobs categories, countries or view user details.
- If the admin choose to manage job categories and countries he can navigate to the those and view, update, delete or add categories and countries.
- If the admin chooses to view user details he can navigate and manage them.
- Admin also can generate reports if necessary.

Consultant sequence diagram


Job Seeker
Job Consultant
Reception
Web-based System
Management
Request consultation appointment
Record appointment request
Confirm appointment request
Check consultant availability
Provide availability information
Display consultant availability to JobSeeker
Provide consultant availability to JobSeeker
Confirm appointment time
Record confirmed appointment
Generate appointment confirmation
Provide appointment confirmation
Update consultant availability
Confirm availability update
Generate "on-demand" reports
Request specific reports
Provide requested reports

Task C

Design Patterns

Design patterns act as a crucial tool in software development because they provide reusable solutions to recurring design challenges. A design pattern is a reusable and generic solution to common recurring issues or obstacles that software engineers face when creating and developing software systems. They act as blueprints for the development of well-structured, maintainable, and efficient software systems. They include best practices, principles, and time-tested approaches for developing software systems.

Design patterns promote code maintainability, scalability, and reusability, making it possible to execute changes and upgrades without disrupting the system. They encourage concern separation, understandable code, and effective communication among development teams. Furthermore, by giving tried-and-true solutions that have been extensively tested and modified through time, these patterns reduce errors.

Singleton design pattern

The Singleton design pattern assures that a class only has one instance and offers global access to that instance. Its main advantage is worldwide access, which enables centralized control of resources such as database connections or configuration settings. When a single instance is sufficient, singleton increases memory efficiency and provides lazy initialization, producing the instance only when it is first requested. A Singleton class, in terms of functionality, often has methods and attributes relevant to the resource it maintains, and it guarantees that all portions of the program interact with the same instance. It may, however, introduce global state and close coupling between components, thus making code more difficult to test and maintain. Furthermore, because to its intrinsic singularity, subclassing Singleton can be difficult.

Factory design pattern

The Factory design pattern provides a degree of abstraction by separating object generation from client code. It encourages abstraction by separating object creation logic from client code, wraps object creation within the factory class, and facilitates extensibility by allowing new product classes to be added without altering existing client code. A Factory class often offers methods for making and returning instances of various product types, insulating client code from object creation details. The Factory design, on the other hand, might introduce additional classes and complexity, especially when handling several factories or product kinds, which can make subclassing factories or dealing with diverse product types difficult.

Abstract factory design pattern

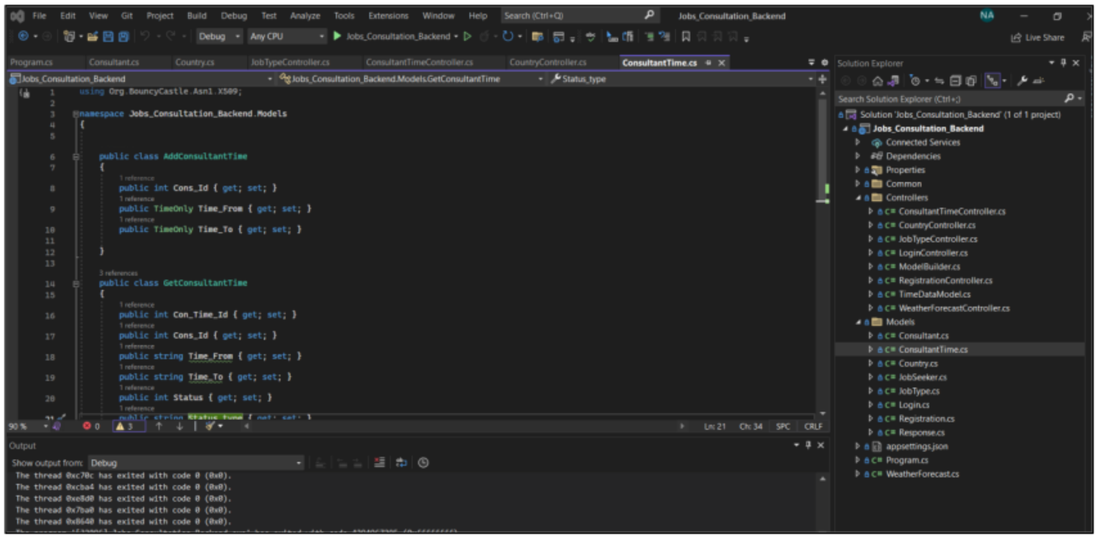
The Abstract Factory design pattern is an interface-based technique to create families of linked items without specifying their specific classes. It encourages object abstraction, encapsulation, and compatibility within the same family. An Abstract Factory specifies techniques for constructing related objects while assuring consistency and family membership. This approach increases flexibility by allowing the switching of whole object families and simplifies maintenance by allowing the inclusion of new product families without modifying current code. It may, however, provide additional complexity, making it more appropriate for sophisticated systems, and changing product families at runtime can be difficult, making it less ideal for dynamic settings.

Design pattern used for developing the system – MVC Design Pattern

Model-View-Controller (MVC) is a key architectural pattern used in software development. Its main feature is the explicit separation of a program into three independent components: Model, View, and Controller. The Model component encompasses the data, business logic, and state management of the application, assuring data integrity and consistency. The View component is in charge of displaying data to users in a user-friendly way, acting as the user interface (UI). The Controller serves as a go-between for the Model and the View, taking user input from the View, analyzing it, and coordinating interactions with the Model to update data or get particular information. This separation of concerns improves the organization, maintainability, and scalability of the code.

The MVC design has several advantages, including explicit separation of concerns, code structure, and maintainability, which increases reusability across diverse contexts. It facilitates successful unit testing, scalable and modular development, and team communication through a shared architectural framework and terminology. However, in smaller applications, the early cost incurred by MVC's structural restrictions should be weighed against the long-term advantages.

Model

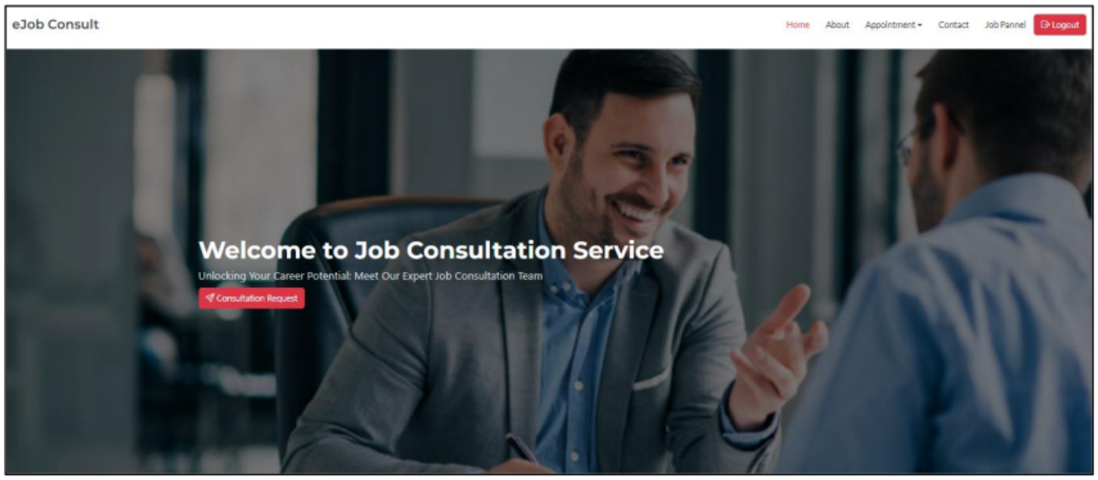


```
using Org.BouncyCastle.Asn1.X509;

namespace Jobs_Consultation_Backend.Models
{


    public class AddConsultantTime
    {
        public int Cons_Id { get; set; }
        public TimeOnly Time_From { get; set; }
        public TimeOnly Time_To { get; set; }

    }

    public class GetConsultantTime
    {
        public int Con_Time_Id { get; set; }
        public int Cons_Id { get; set; }
        public string Time_From { get; set; }
        public string Time_To { get; set; }
        public int Status { get; set; }
        public string Status_type { get; set; }
```

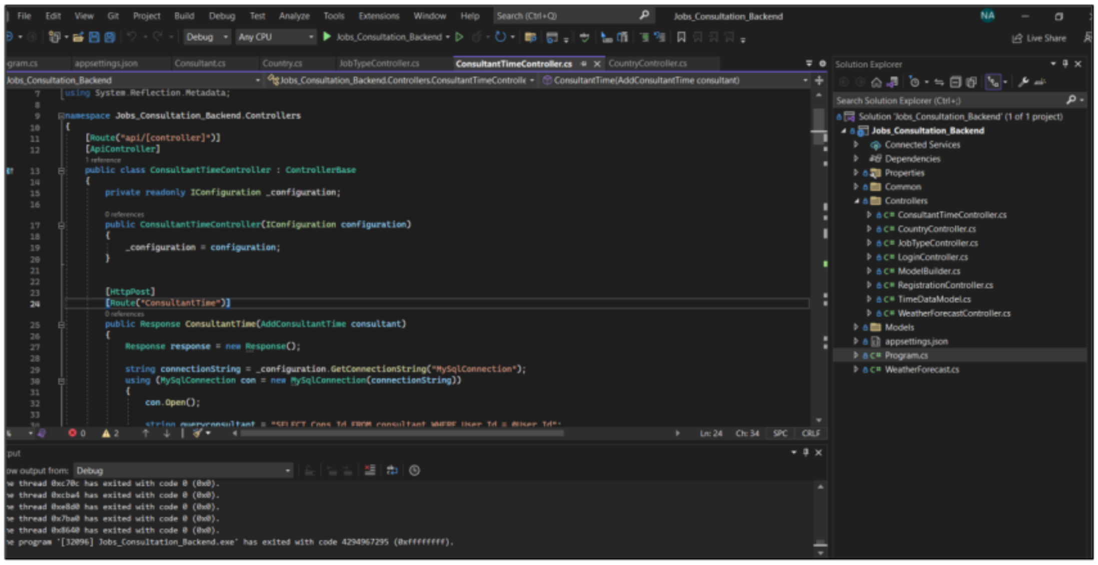
```
The thread 0xc70c has exited with code 0 (0x0).
The thread 0xcba4 has exited with code 0 (0x0).
The thread 0xe8d0 has exited with code 0 (0x0).
The thread 0x7ba0 has exited with code 0 (0x0).
The thread 0x8640 has exited with code 0 (0x0).
```

```
Solution 'Jobs_Consultation_Backend' (1 of 1 project)
  Jobs_Consultation_Backend
    Connected Services
    Dependencies
    Properties
    Common
    Controllers
      ConsultantTimeController.cs
      CountryController.cs
      JobTypeController.cs
      LoginController.cs
      ModelBuilder.cs
      RegistrationController.cs
      TimeDataModel.cs
      WeatherForecastController.cs
    Models
      Consultant.cs
      ConsultantTime.cs
      Country.cs
      JobSeeker.cs
      JobType.cs
      Login.cs
      Registration.cs
      Response.cs
    appsettings.json
    Program.cs
    WeatherForecast.cs
```

Views


eJob Consult
Home
About
Appointment
Contact
Job Pannel
Logout
Welcome to Job Consultation Service
Unlocking Your Career Potential: Meet Our Expert Job Consultation Team
Consultation Request

Controller



```
using System.Reflection.Metadata;

namespace Jobs_Consultation_Backend.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ConsultantTimeController : ControllerBase
    {
        private readonly IConfiguration _configuration;

        public ConsultantTimeController(IConfiguration configuration)
        {
            _configuration = configuration;
        }


        [HttpPost]
        [Route("ConsultantTime")]
        public Response ConsultantTime(AddConsultantTime consultant)
        {
            Response response = new Response();

            string connectionString = _configuration.GetConnectionString("MySqlConnection");
            using (MySqlConnection con = new MySqlConnection(connectionString))
            {
                con.Open();

                string queryconsultant = "SELECT Cons_Id FROM consultant WHERE User_Id = @User_Id";
```

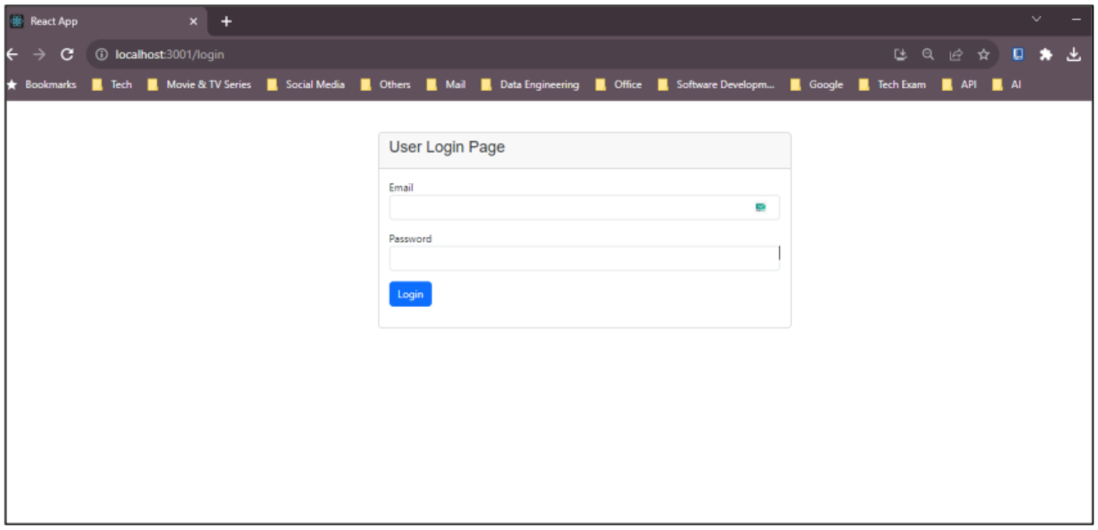
```
ne thread 0xc70c has exited with code 0 (0x0).
ne thread 0xcba4 has exited with code 0 (0x0).
ne thread 0xe8d0 has exited with code 0 (0x0).
ne thread 0x7ba0 has exited with code 0 (0x0).
ne thread 0x8640 has exited with code 0 (0x0).
ne program '[32096] Jobs_Consultation_Backend.exe' has exited with code 4294967295 (0xffffffff).
```

Using the MVC (Model-View-Controller) architecture pattern for the built consultation center system is a wise decision with numerous significant benefits. MVC's distinct separation of concerns guarantees that the system is both manageable and extendable. It divides data management (Model), user interface (View), and user interaction (Controller), resulting in better code structure and readability. This flexibility enables separate development teams or developers to work on discrete components concurrently, hence speeding up the development process. Furthermore, by segregating components for unit testing, MVC simplifies testing.

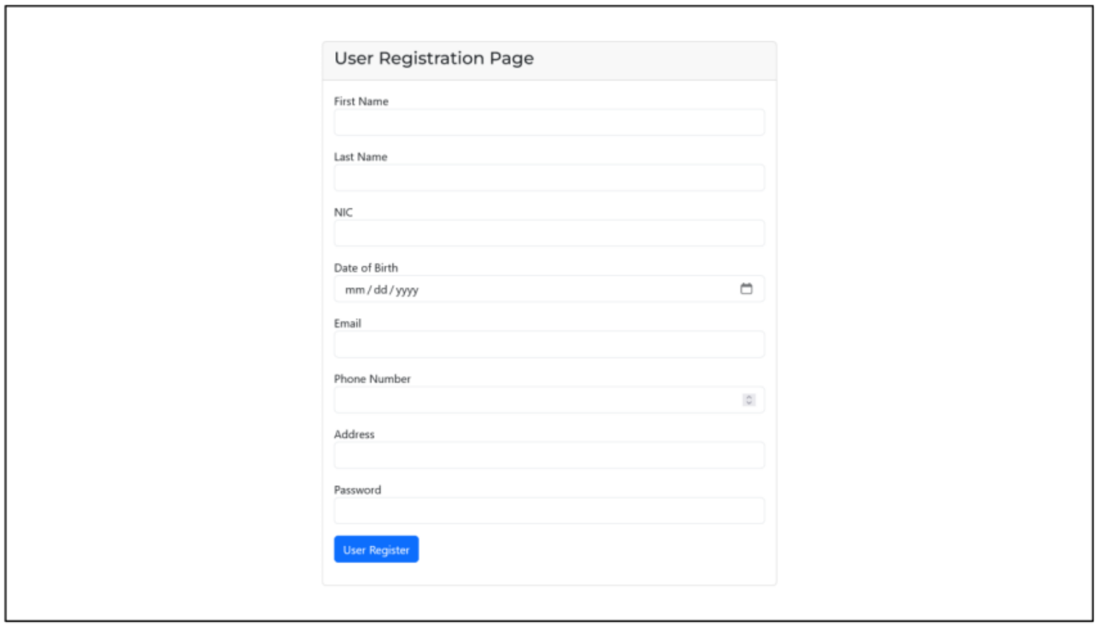
Tasks D

User Registration and Login

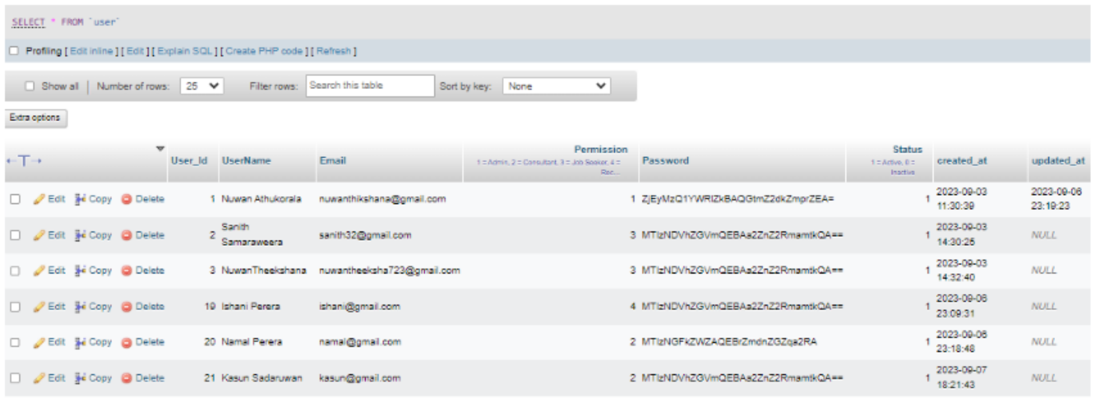
Login Page



Registration Page



User Table In DB



```
SELECT * FROM `user`
```

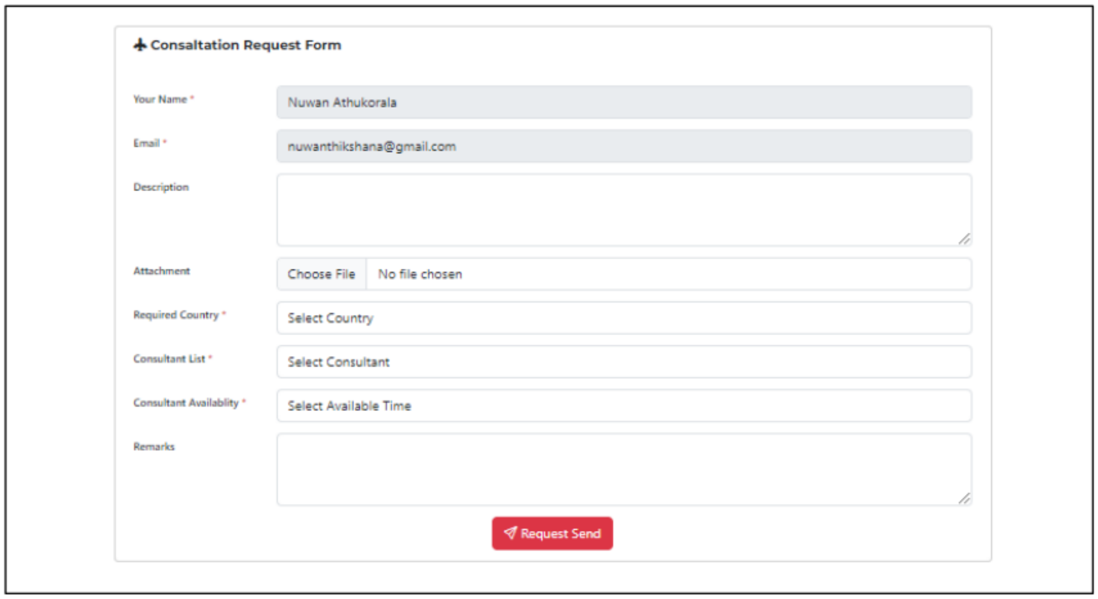
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	User_Id	UserName	Email	Permission 1 = Admin, 2 = Consultant, 3 = Job Seeker, 4 = Rec...	Password	Status 1 = Active, 0 = Inactive	created_at	updated_at
Edit Copy Delete	1	Nuwan Athukorala	nuwanthikshana@gmail.com	1	ZjEyMzQ1YWRlZkBAQGtmZ2dkZmprZEA=	1	2023-09-03 11:30:39	2023-09-06 23:19:23
Edit Copy Delete	2	Sanith Samaraweera	sanith32@gmail.com	3	MTIzNDVhZGVmQEBAa2ZnZ2RmamtkQA==	1	2023-09-03 14:30:26	NULL
Edit Copy Delete	3	NuwanTheekshana	nuwantheeksha723@gmail.com	3	MTIzNDVhZGVmQEBAa2ZnZ2RmamtkQA==	1	2023-09-03 14:32:40	NULL
Edit Copy Delete	19	Ishani Perera	ishani@gmail.com	4	MTIzNDVhZGVmQEBAa2ZnZ2RmamtkQA==	1	2023-09-06 23:09:31	NULL
Edit Copy Delete	20	Namal Perera	namal@gmail.com	2	MTIzNGFkZWZAQEBrZmdnZGZqa2RA	1	2023-09-06 23:18:48	NULL
Edit Copy Delete	21	Kasun Sadaruwan	kasun@gmail.com	2	MTIzNDVhZGVmQEBAa2ZnZ2RmamtkQA==	1	2023-09-07 18:21:43	NULL

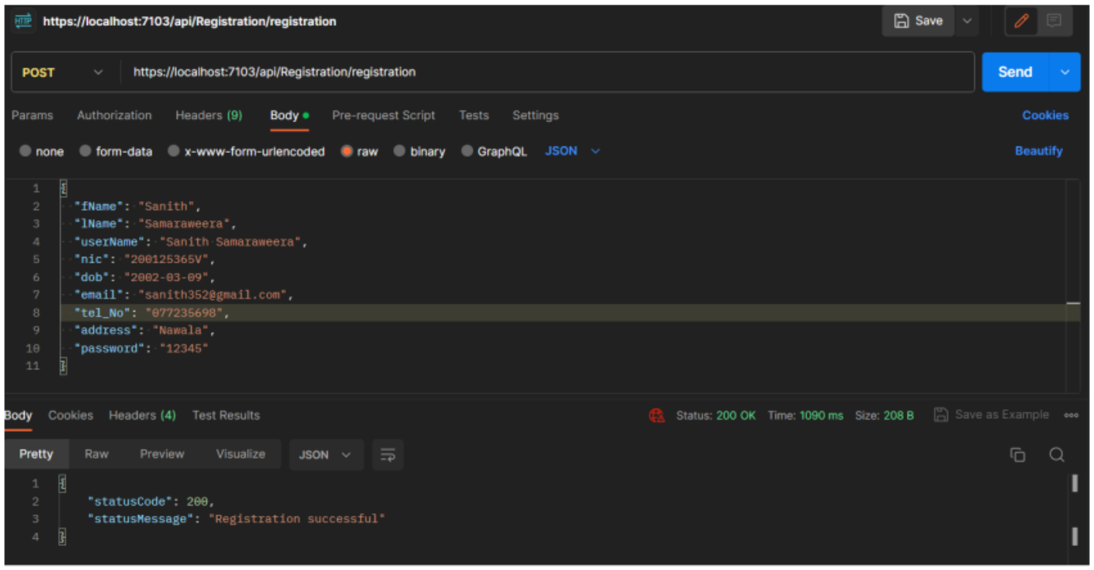
Consultation Appointment Scheduling



Consaltation Request Form

Your Name *	Nuwan Athukorala
Email *	nuwanthikshana@gmail.com
Description	
Attachment	Choose File No file chosen
Required Country *	Select Country
Consultant List *	Select Consultant
Consultant Availablity *	Select Available Time
Remarks	

Web Services Implementation



https://localhost:7103/api/Registration/registration

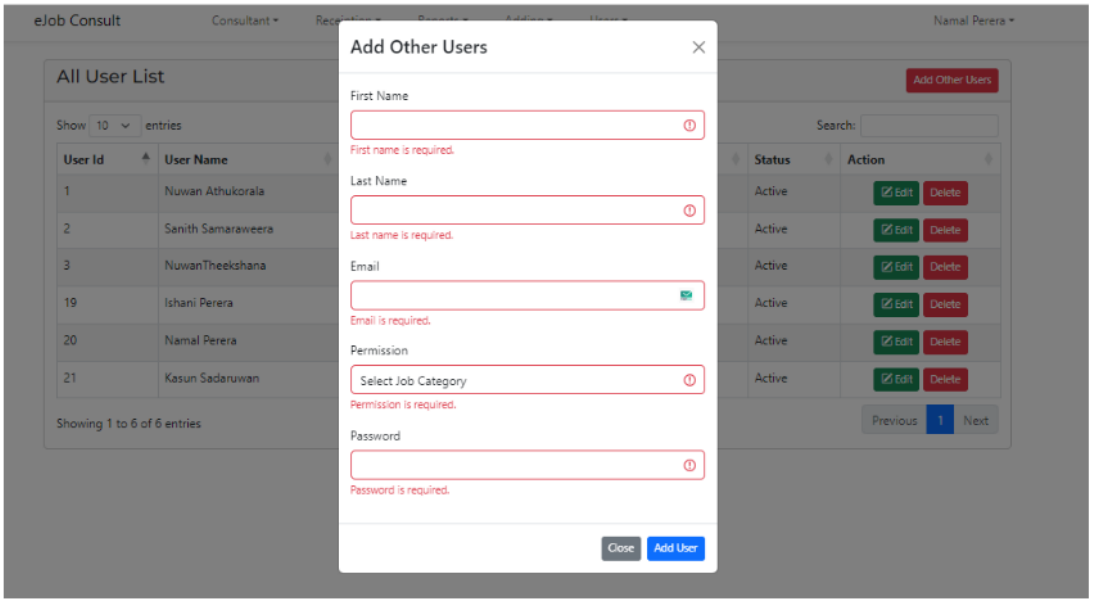
POST https://localhost:7103/api/Registration/registration

```
{
"fName": "Sanith",
"lName": "Samaraweera",
"userName": "Sanith Samaraweera",
"nic": "200125365V",
"dob": "2002-03-09",
"email": "sanith352@gmail.com",
"tel_No": "077235698",
"address": "Nawala",
"password": "12345"
}
```

Status: 200 OK Time: 1090 ms Size: 208 B

```
{
    "statusCode": 200,
    "statusMessage": "Registration successful"
}
```

Input Validation



eJob Consult

All User List

Show 10 entries

User Id	User Name	Status	Action
1	Nuwan Athukorala	Active	Edit Delete
2	Sanith Samaraweera	Active	Edit Delete
3	NuwanTheekshana	Active	Edit Delete
19	Ishani Perera	Active	Edit Delete
20	Namal Perera	Active	Edit Delete
21	Kasun Sadaruwan	Active	Edit Delete

Showing 1 to 6 of 6 entries

Add Other Users

First Name

First name is required.

Last Name

Last name is required.

Email

Email is required.

Permission

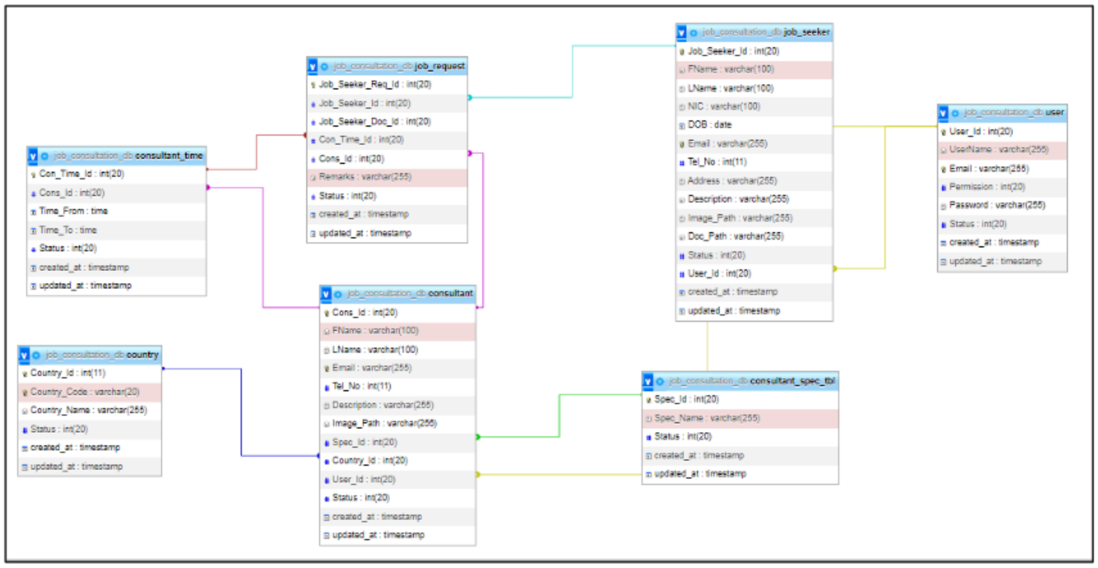
Select Job Category

Permission is required.

Password

Password is required.

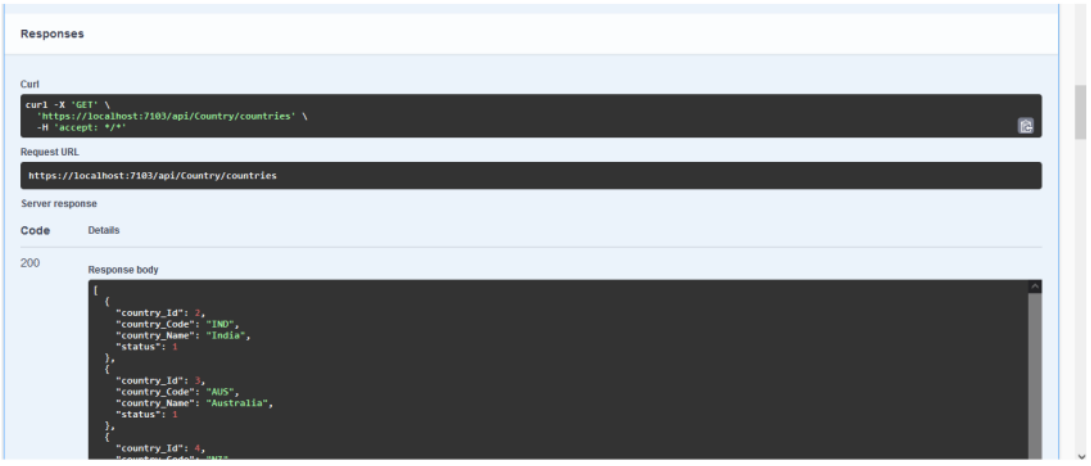
Database Integration



Tasks E

API Testing

GET


Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7103/api/Country/countries' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7103/api/Country/countries
```

Server response

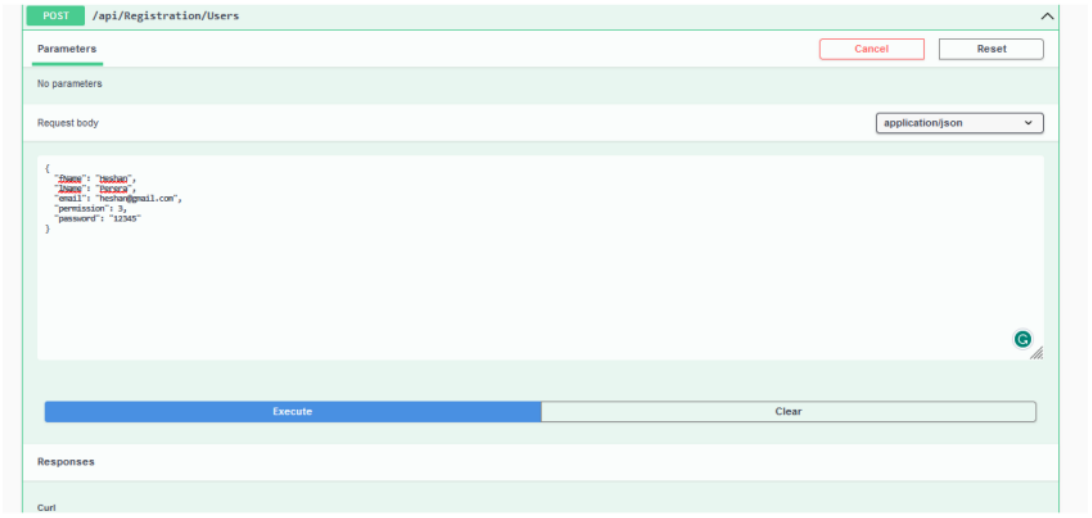
Code Details

200

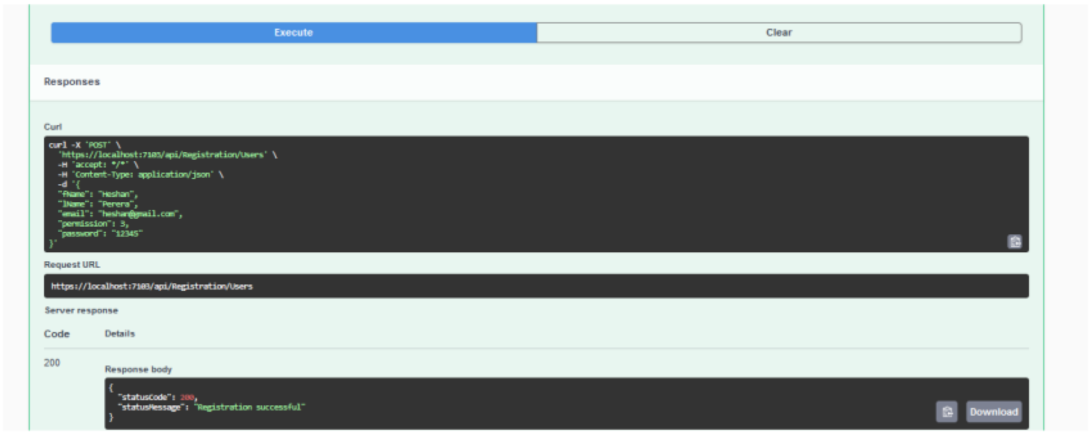
Response body

```
[
  {
    "country_Id": 2,
    "country_Code": "IND",
    "country_Name": "India",
    "status": 1
  },
  {
    "country_Id": 3,
    "country_Code": "AUS",
    "country_Name": "Australia",
    "status": 1
  },
  {
    "country_Id": 4,
```

POST



```
{
  "fName": "Heshan",
  "lName": "Perera",
  "email": "heshan@gmail.com",
  "permission": 3,
  "password": "12345"
}
```

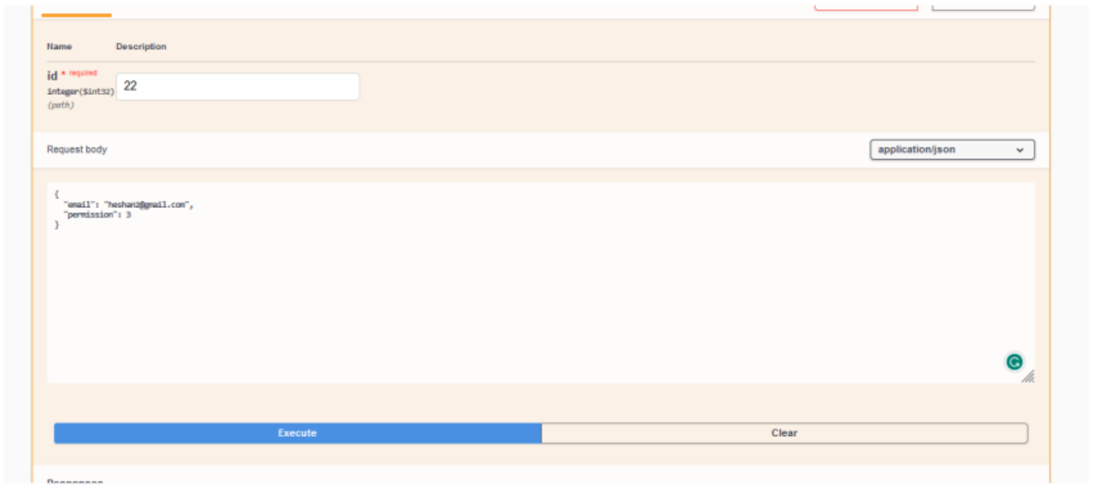


```
curl -X 'POST' \
  'https://localhost:7103/api/Registration/Users' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "fName": "Heshan",
  "lName": "Perera",
  "email": "heshan@gmail.com",
  "permission": 3,
  "password": "12345"
}'
```

```
https://localhost:7103/api/Registration/Users
```

```
{
  "statusCode": 200,
  "statusMessage": "Registration successful"
}
```

PUT

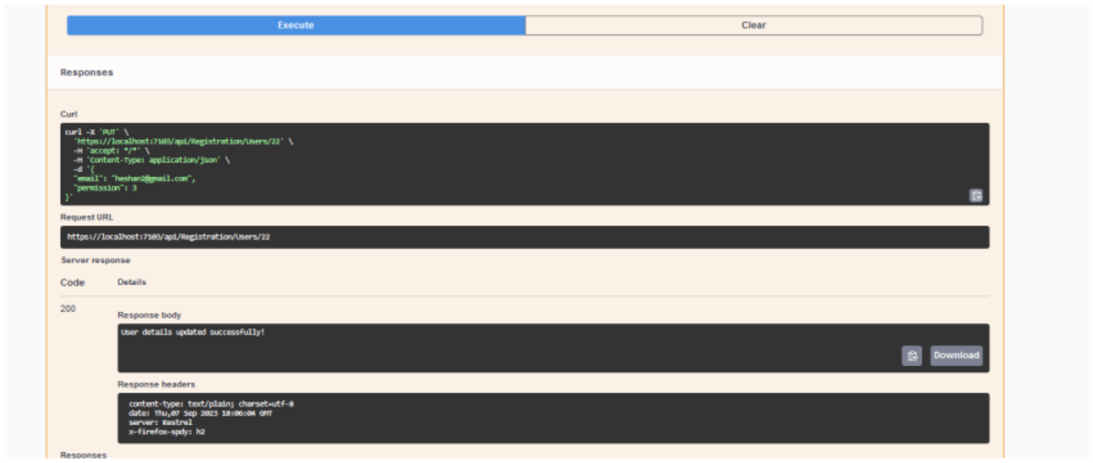


Name Description

id * required integer($int32) (path): 22

Request body: application/json

```
{
  "email": "heshan2@gmail.com",
  "permission": 3
}
```



Curl

```
curl -X 'PUT' \
  'https://localhost:7103/api/Registration/Users/22' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "heshan2@gmail.com",
  "permission": 3
}'
```

Request URL

```
https://localhost:7103/api/Registration/Users/22
```

Server response

Code: 200

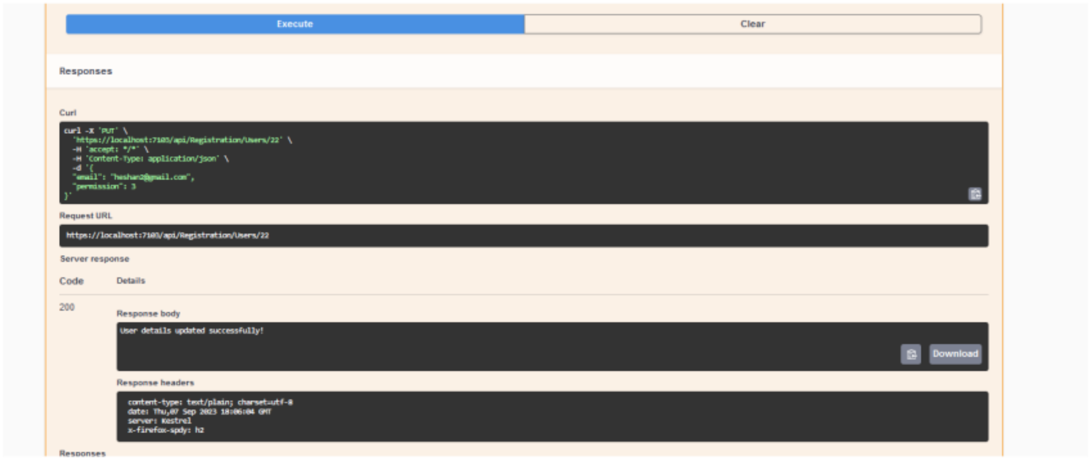
Response body

```
User details updated successfully!
```

Response headers

```
 content-type: text/plain; charset=utf-8
 date: Thu,07 Sep 2023 18:06:04 GMT
 server: Kestrel
 x-firefox-spdy: h2
```

DELETE



```
curl -X 'PUT' \
  'https://localhost:7103/api/Registration/Users/22' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "heshan2@gmail.com",
  "permission": 3
}'
```

```
https://localhost:7103/api/Registration/Users/22
```

```
User details updated successfully!
```

```
content-type: text/plain; charset=utf-8
date: Thu,07 Sep 2023 18:06:04 GMT
server: Kestrel
x-firefox-spdy: h2
```

Task F

User Documentation

1. **Registration**
2. **Login**
3. **Consultant Request Send**
4. **Consultant Accept Pannel**
5. **Report Page**
6. **Manage Consultant Times**
7. **Jobseeker Registration**
8. **Consultant Registration**
9. **Other User Registration (Admin / Reception)**

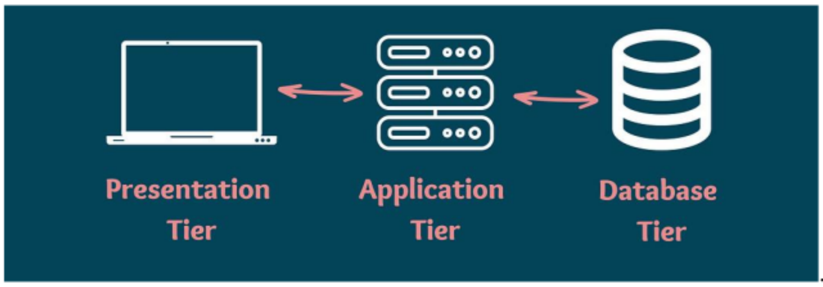
Technical Documentation

Introduction

The system's goal is to establish an effective and user-friendly platform for managing job consultants, job searchers, and appointments at the Jobs Consultation Centre. This system is intended to replace the current manual paper-based approach by streamlining the process of appointment scheduling.

System Architecture

System developed using MVC architecture and a three tier distributed architecture. MVC uses Model, view and controller to develop the system. MVC makes it easier to manage and maintain the web application and work on multiple components



Technologies used

- Frontend – React
- Backend – ASP.Net core 6.0
- Database Schema – MySQL Database
- Framework – Bootstrap
- Programming languages – C#, JavaScript, HTML, CSS

Web Services

RESTful API which act as an interface between two applications that uses HTTP requests to access data, was used in development of the application.

Methods – GET , POST, PUT, DELETE

Security Measures

- Password Encryption
- Access Control
- CRFS Tokens
- Authentication methods

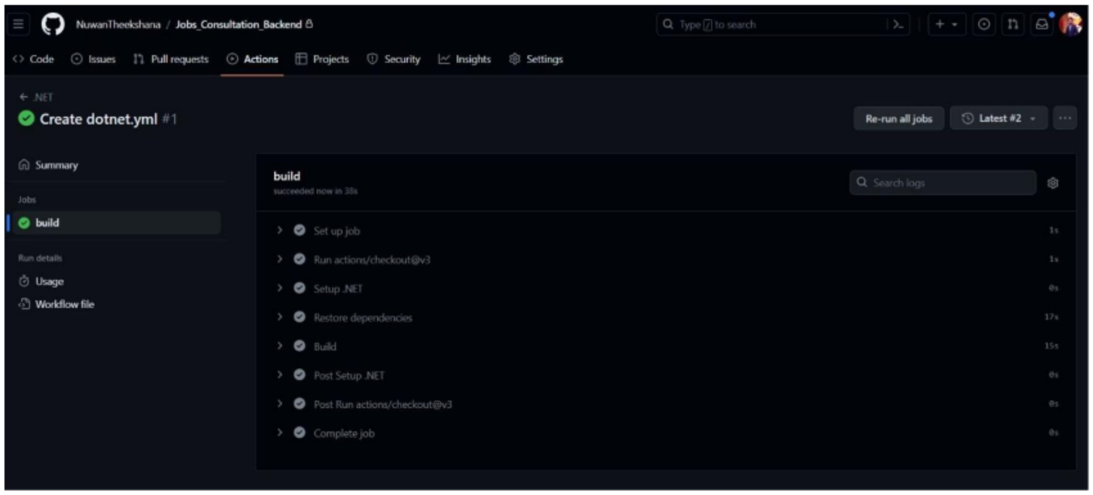
Maintenance and system updates

- Version controlling available using GitHub
- System backup available as a GitHub Repo
- Database backup

Task G

Git Version control

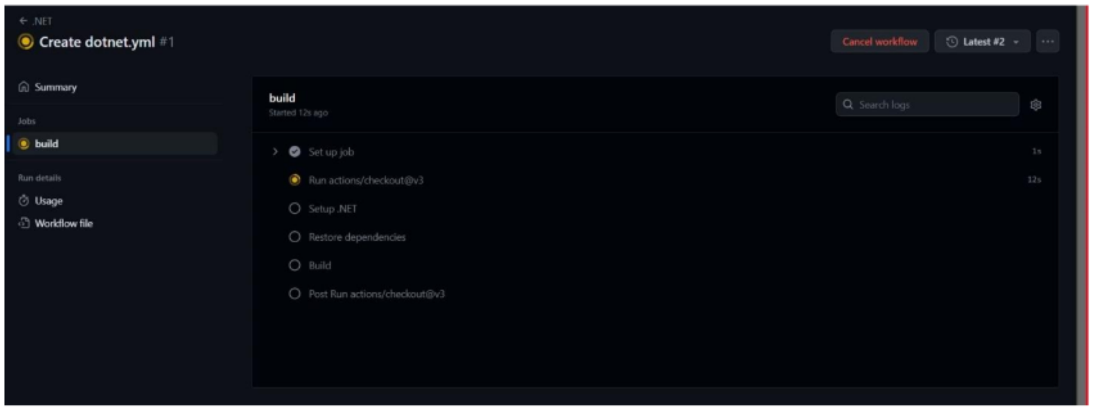
Git is a distributed version control system that allows developers to monitor changes, collaborate, and manage source code effectively. Git allows collaborative coding and project management with capabilities such as branching, merging, and a staging space for exact commits. Its open-source nature, solid security, and broad tool ecosystem make it the chosen choice for people, open-source communities, and companies throughout the world, assuring code integrity and smooth development workflows. In the process of developing the system of Jobs Consultation Center regular changes were uploaded to GitHub utilizing version control techniques.



← .NET
Create dotnet.yml #1

build
succeeded now in 38s

```
Set up job                      1s
Run actions/checkout@v3         1s
Setup .NET                      0s
Restore dependencies           17s
Build                          15s
Post Setup .NET                 0s
Post Run actions/checkout@v3    0s
Complete job                    0s
```



← .NET
Create dotnet.yml #1

build
Started 12s ago

```
Set up job                      1s
Run actions/checkout@v3        12s
Setup .NET
Restore dependencies
Build
Post Run actions/checkout@v3
```

Nuwan Athukorala - 20284580.docx

ORIGINALITY REPORT

10%	2%	0%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Wales Institute, Cardiff Student Paper	6%
2	Submitted to The University of the South Pacific Student Paper	2%
3	Submitted to Academy of Information Technology Student Paper	1%
4	Submitted to Bahcesehir University Student Paper	1%
5	Submitted to Central Queensland University Student Paper	1%
6	commerce.financialpost.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		