Blazor Web Documentation

BR20-2

2021

Foreword

Thank you for granting me the opportunity to be a candidate to join the Research and Development team in Software Laboratory Center at Binus University. Thank you for everyone at Operation Division for leading me throughout the semester, everyone at Research and Development for choosing me to be a candidate, and everyone at Software Laboratory Center who has helped and accompanied me to reach the state I am currently at, especially my generation (20-2). I am grateful and will try my best at doing this test.

Technologies

ASP.NET Core 5.0

ASP.NET Core 5.0 is the latest web framework for developing web applications on the .NET platform by Microsoft. It is better from its predecessor, ASP.NET 4.x, as ASP.NET Core is much more performant, provides cross platform solution, and its latest UI technology, Blazor.

ASP.NET Core Identity

ASP.NET Core Identity provides authentication and authorization scaffolding to ASP.NET Core projects with built-in supports for login register, social login, database migration, and more.

Blazor (Server)

Blazor is part of ASP.NET to build interactive web UI with C#. There are two hosting models for Blazor, Blazor WebAssembly and Blazor Server. Blazor Server is used in this application to lighten up client's load and prerender pages server-side before sending them to client.

Entity Framework Core + SQL Server

Entity Framework Core is used as ORM in this application together with SQL Server to help in writing data-access code and optimizing database queries.

Bootstrap 4.3.1

ASP.NET Core project template comes with Bootstrap 4.3.1 out of the box to enable rapid styling.

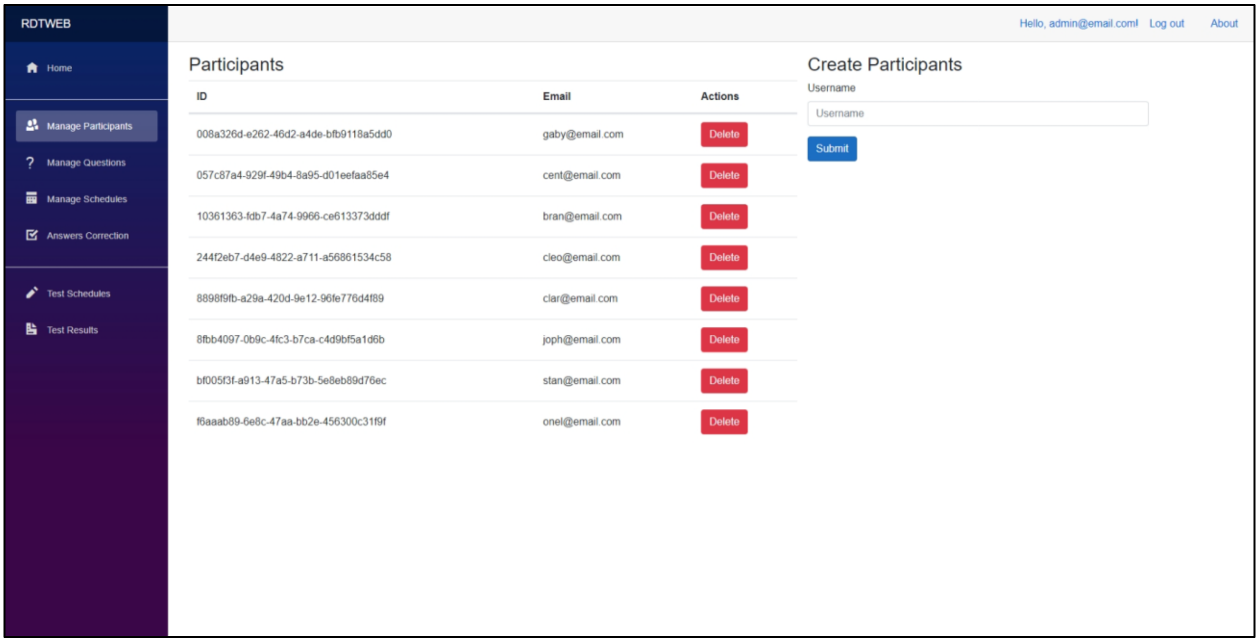
Pages

All pages are placed under “Pages” folder.

Admin

All pages under “Admin” folder are authorized for users with admin role only.

Manage Participants



Participants

ID	Email	Actions
008a326d-e262-46d2-a4de-bfb9118a5dd0	gaby@email.com	Delete
057c87a4-929f-49b4-8a95-d01eefaa85e4	cent@email.com	Delete
10361363-fdb7-4a74-9966-ce613373dddf	bran@email.com	Delete
244f2eb7-d4e9-4822-a711-a56861534c58	cleo@email.com	Delete
8898f9fb-a29a-420d-9e12-96fe776d4f89	clar@email.com	Delete
8fbb4097-0b9c-4fc3-b7ca-c4d9bf5a1d6b	joph@email.com	Delete
bf005f3f-a913-47a5-b73b-5e8eb89d76ec	stan@email.com	Delete
f6aaab89-6e8c-47aa-bb2e-456300c31f9f	onel@email.com	Delete

Create Participants

Username

Username

Submit

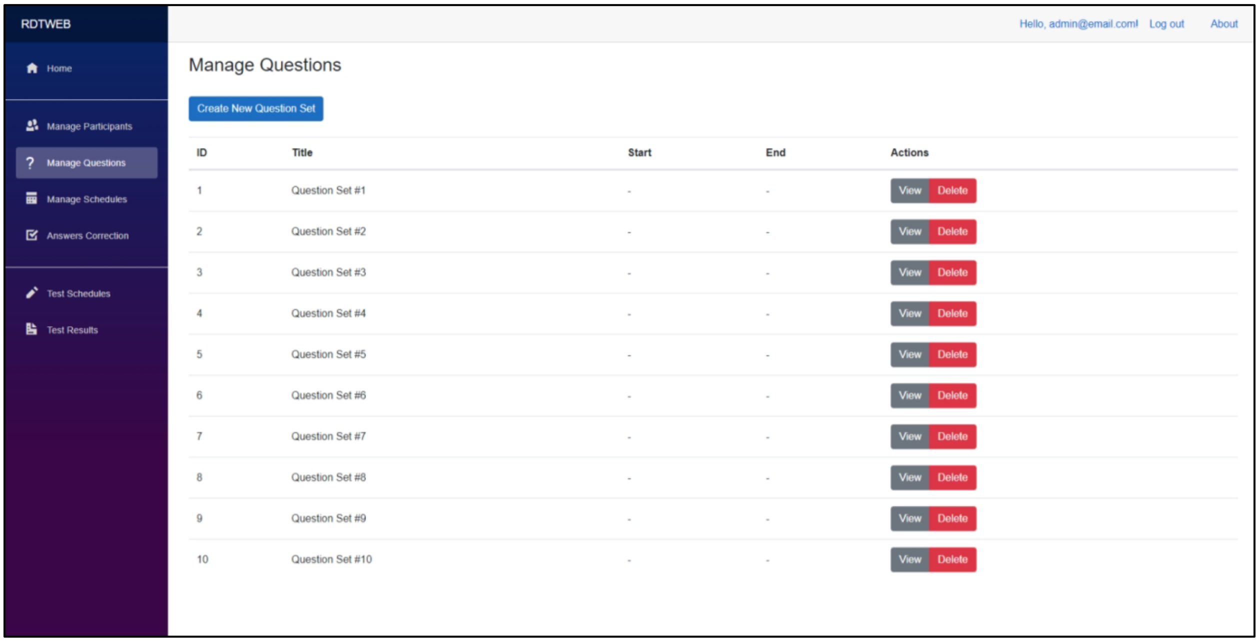
Figure 1. Manage participants.

Admin can view all participants, create, and delete participants in this page.

Manage Questions

Admin can view all questions, view question detail, update and delete questions in this page. Manage Questions are divided into three pages: Index, Create, and View.

Index



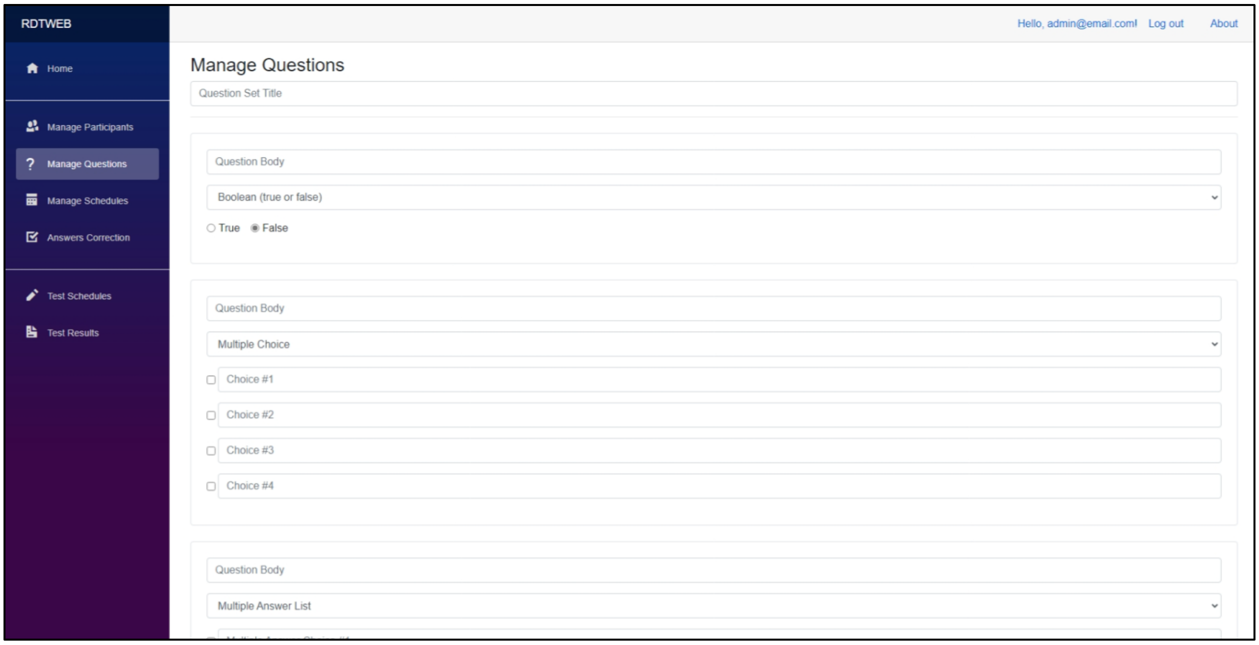
Manage Questions

ID	Title	Start	End	Actions
1	Question Set #1	-	-	View Delete
2	Question Set #2	-	-	View Delete
3	Question Set #3	-	-	View Delete
4	Question Set #4	-	-	View Delete
5	Question Set #5	-	-	View Delete
6	Question Set #6	-	-	View Delete
7	Question Set #7	-	-	View Delete
8	Question Set #8	-	-	View Delete
9	Question Set #9	-	-	View Delete
10	Question Set #10	-	-	View Delete

Figure 2. Manage questions index.

This page shows all question sets. Question sets can be deleted.

Create



Manage Questions

Question Set Title

Question Body

Boolean (true or false)

○ True ● False

Question Body

Multiple Choice

□ Choice #1

□ Choice #2

□ Choice #3

□ Choice #4

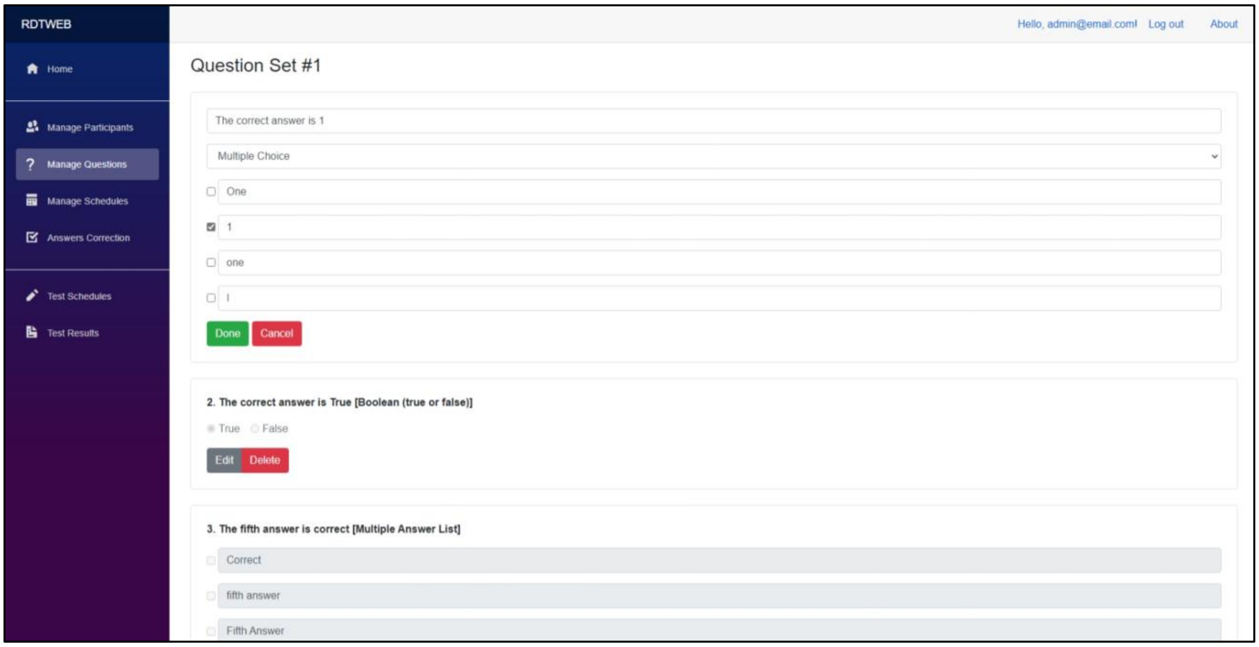
Question Body

Multiple Answer List

Figure 3. Create question set.

This page shows options for creating question set. Admin is free to add types of question and number of questions in a question set. Admin can also set question's correct answer.

View



Question Set #1

The correct answer is 1

Multiple Choice

□ One

☑ 1

□ one

□ I

2. The correct answer is True [Boolean (true or false)]

● True ○ False

3. The fifth answer is correct [Multiple Answer List]

□ Correct

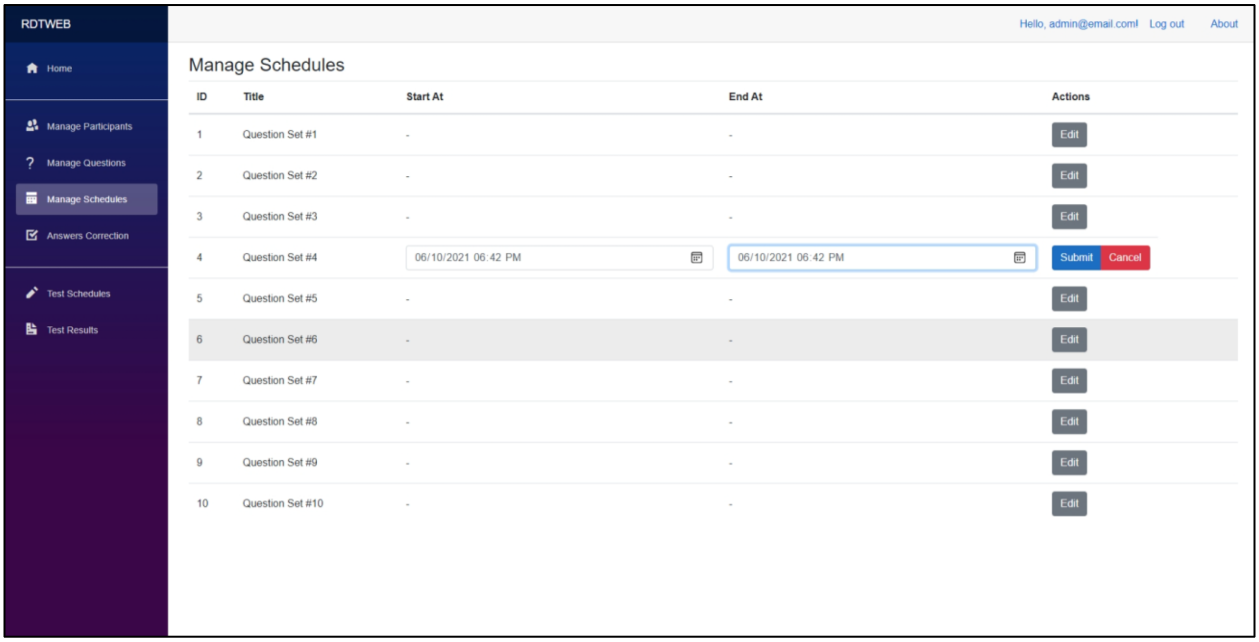
□ fifth answer

□ Fifth Answer

Figure 4. View question set detail.

Admin can view all questions in a question set, update its type, and correct answer, and delete questions.

Manage Schedules



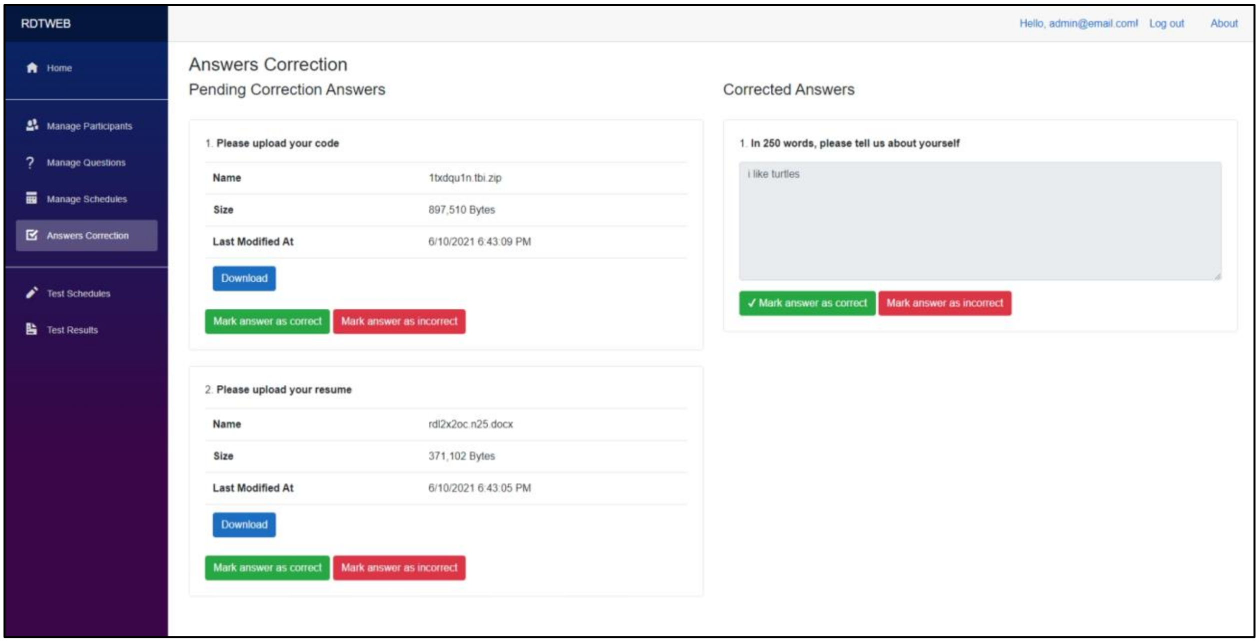
Manage Schedules

ID	Title	Start At	End At	Actions
1	Question Set #1	-	-	Edit
2	Question Set #2	-	-	Edit
3	Question Set #3	-	-	Edit
4	Question Set #4	06/10/2021 06:42 PM	06/10/2021 06:42 PM	Submit Cancel
5	Question Set #5	-	-	Edit
6	Question Set #6	-	-	Edit
7	Question Set #7	-	-	Edit
8	Question Set #8	-	-	Edit
9	Question Set #9	-	-	Edit
10	Question Set #10	-	-	Edit

Figure 5. Manage schedules.

Admin can edit start and end time for each question sets listed in this page.

Answer Corrections



Answers Correction

Pending Correction Answers

1. **Please upload your code**

Name	1txdqu1n.tbi.zip
Size	897,510 Bytes
Last Modified At	6/10/2021 6:43:09 PM

2. **Please upload your resume**

Name	rdl2x2oc.n25.docx
Size	371,102 Bytes
Last Modified At	6/10/2021 6:43:05 PM

Corrected Answers

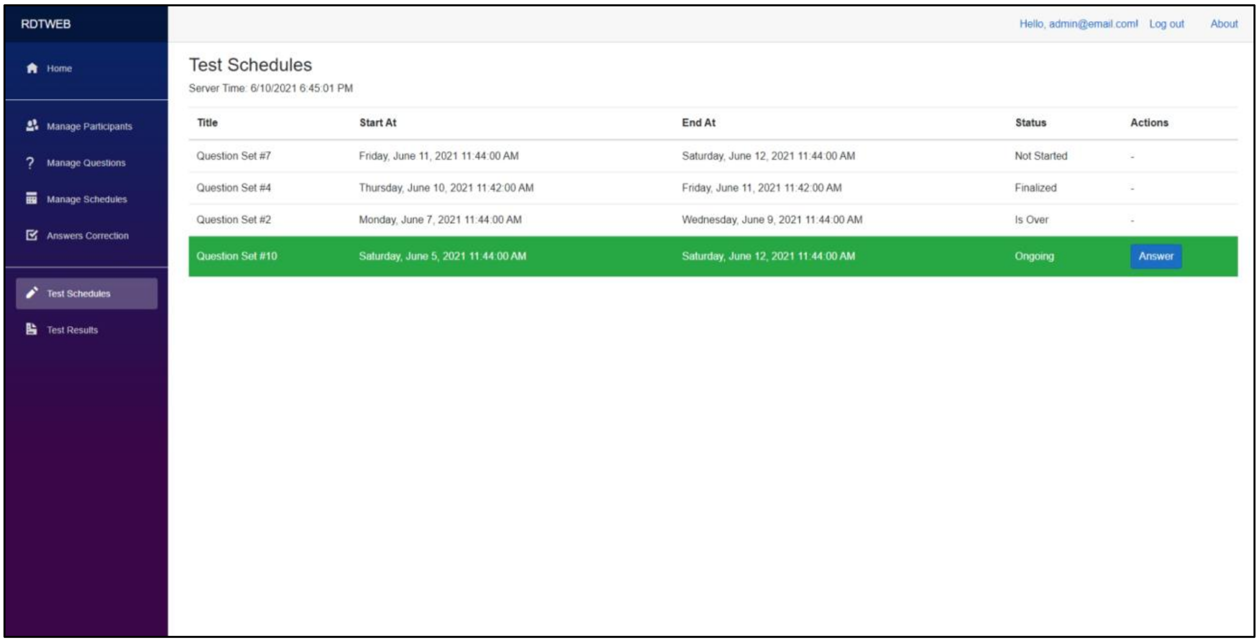
1. **In 250 words, please tell us about yourself**

i like turtles

Figure 6. Answer corrections.

Admin can mark an answer to be either correct or incorrect. Questions shown here are only of type “Essay” and “Submit File”.

Test Schedules



Test Schedules

Server Time: 6/10/2021 6:45:01 PM

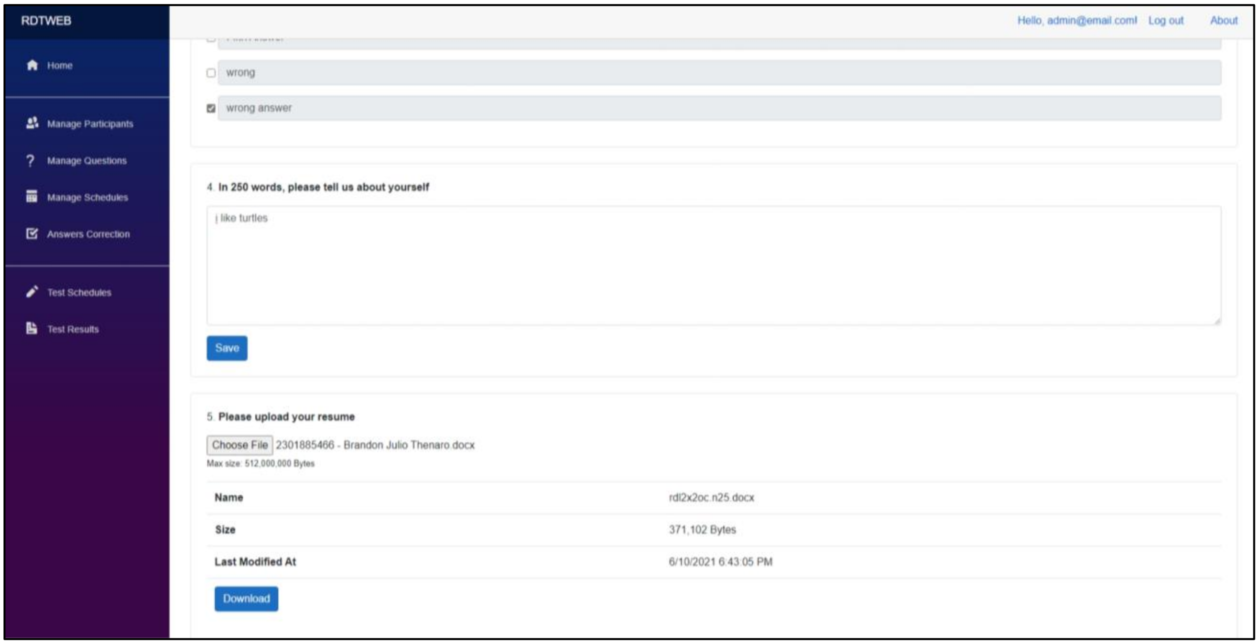
Title	Start At	End At	Status	Actions
Question Set #7	Friday, June 11, 2021 11:44:00 AM	Saturday, June 12, 2021 11:44:00 AM	Not Started	-
Question Set #4	Thursday, June 10, 2021 11:42:00 AM	Friday, June 11, 2021 11:42:00 AM	Finalized	-
Question Set #2	Monday, June 7, 2021 11:44:00 AM	Wednesday, June 9, 2021 11:44:00 AM	Is Over	-
Question Set #10	Saturday, June 5, 2021 11:44:00 AM	Saturday, June 12, 2021 11:44:00 AM	Ongoing	Answer

Figure 7. Test schedules.

Users can view all question sets that are either one of “Not Started”, “Ongoing”, “Is Over”, or “Finalized”. Users can attempt to answer an “Ongoing” question set and be taken to “Test Answer” page. Below explains about the test statuses:

- “Finalized” is when the user has finalized their answers in the test.
- “Not started” is when the start date is later than now.
- “Ongoing” is when now is later than start date and now is former than end date.
- “Is Over” is when now is later then end date.

Test Answer



□ wrong

☑ wrong answer

4. **In 250 words, please tell us about yourself**

i like turtles

5. **Please upload your resume**

2301885466 - Brandon Julio Thenaro.docx

Max size: 512,000,000 Bytes

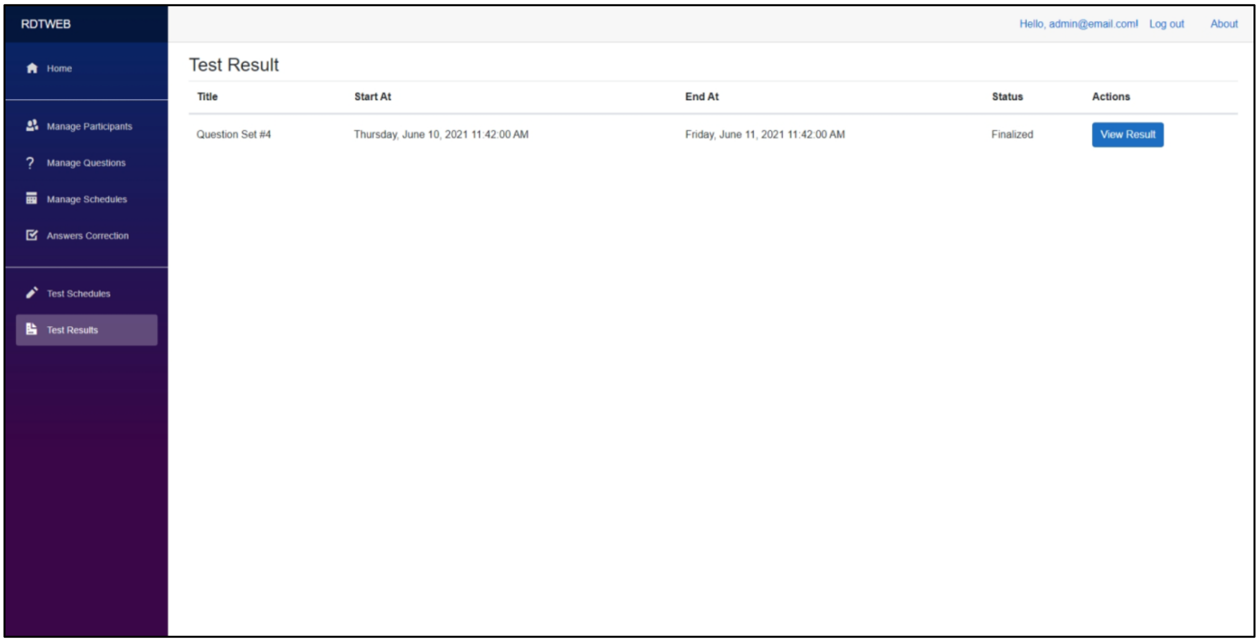
Name	rdl2x2oc.n25.docx
Size	371,102 Bytes
Last Modified At	6/10/2021 6:43:05 PM

Figure 8. Test answer.

Users can answer a question set in this page. If a user has finalized their answer for the selected question set, then the user cannot update their answers anymore.

Test Result

Index



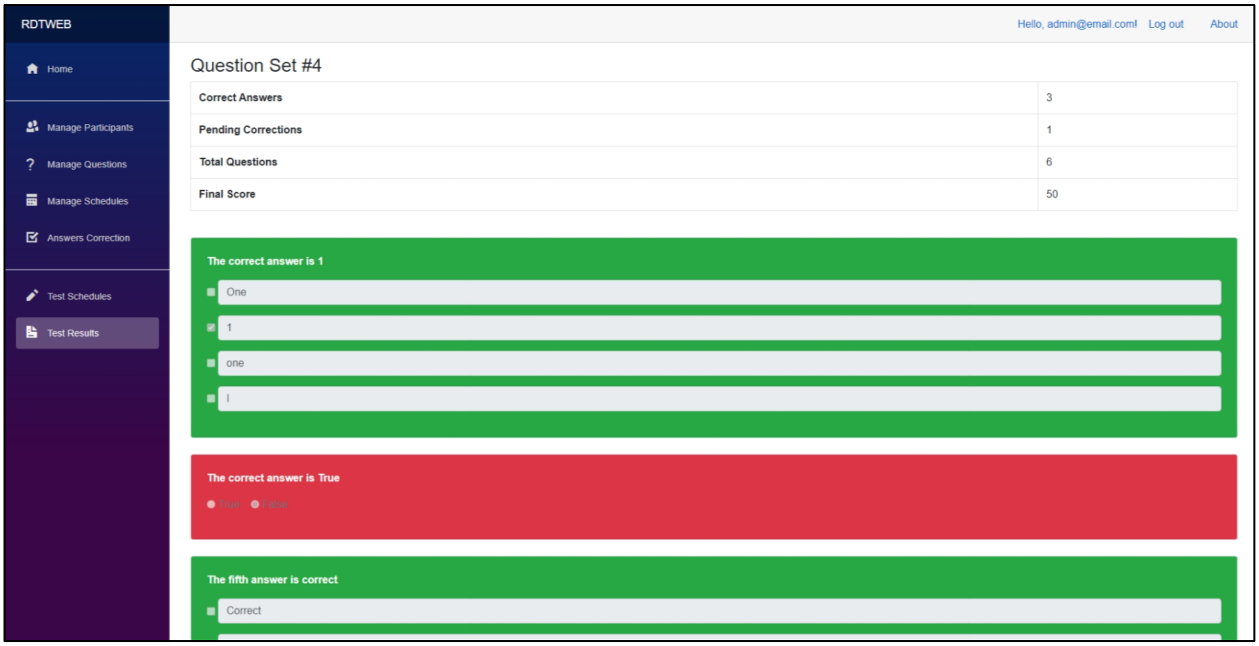
Test Result

Title	Start At	End At	Status	Actions
Question Set #4	Thursday, June 10, 2021 11:42:00 AM	Friday, June 11, 2021 11:42:00 AM	Finalized	View Result

Figure 9. Test result index.

User can view question sets that they already answered and finalized. If the user has not finalized their answers, they will not be shown in this page.

View



Question Set #4

Correct Answers	3
Pending Corrections	1
Total Questions	6
Final Score	50

The correct answer is 1

☐ One
☑ 1
☐ one
☐ I

The correct answer is True

○ True ● False

The fifth answer is correct

☐ Correct

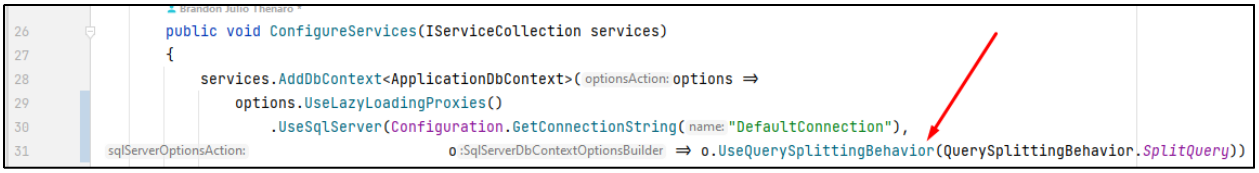
Figure 10. Test result view.

User can view their answers in a question set and view their answer status, that is either correct, incorrect, or pending correction. Grading information is also presented in this page.

Performance Consideration

Split Query

Entity Framework Core’s split query[1] feature is used to prevent “cartesian explosion[2]” problem when querying entities with one-to-many relationship. Hence, the solution is to split the query. The application enables split query globally to fix this problem in “Startup.cs”.

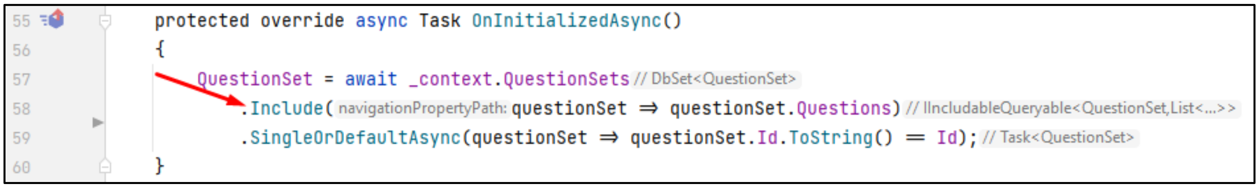


```
Brandon Julio Thenaro *
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>( optionsAction: options ⇒
        options.UseLazyLoadingProxies()
            .UseSqlServer(Configuration.GetConnectionString( name: "DefaultConnection"),
sqlServerOptionsAction:                o :SqlServerDbContextOptionsBuilder  ⇒ o.UseQuerySplittingBehavior(QuerySplittingBehavior.SplitQuery))
```

Figure 11. Split query configuration in "Startup.cs".

Eager Loading

Relationships are eagerly loaded and explicitly specified to only load data which are required. Lazy loading is also used but sparingly to prevent N+1 problem, e.g., while looping[3].



```
protected override async Task OnInitializedAsync()
{
    QuestionSet = await _context.QuestionSets // DbSet<QuestionSet>
        .Include( navigationPropertyPath: questionSet ⇒ questionSet.Questions) // IIncludableQueryable<QuestionSet,List<...>>
        .SingleOrDefaultAsync(questionSet ⇒ questionSet.Id.ToString() ═ Id); // Task<QuestionSet>
}
```

Figure 12. An example of explicit eager loading in” ManageQuestions/Index.razor”.

Virtualize Component

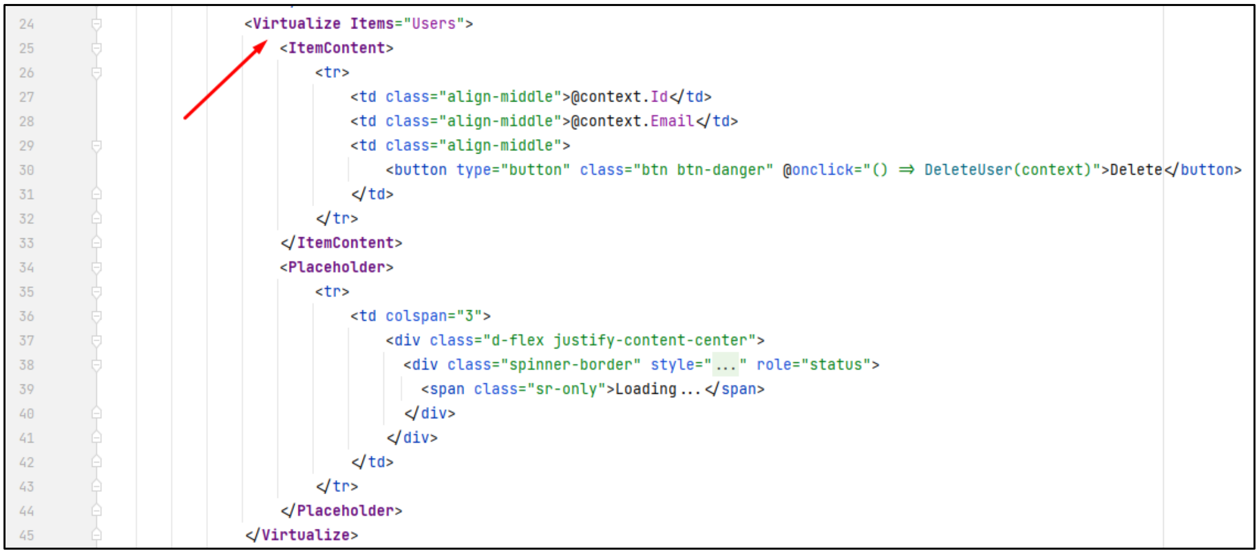
Virtualize component is built-in Blazor component to improve UI performance while rendering list by limiting UI rendering only to parts that are currently visible[4].

[1] Single vs. Split Queries - EF Core | Microsoft Docs

[2] The performance issue "Cartesian Explosion" made its comeback in Entity Framework Core 3 - Thinktecture

[3] Efficient Querying - EF Core | Microsoft Docs

[4] ASP.NET Core Blazor component virtualization | Microsoft Docs

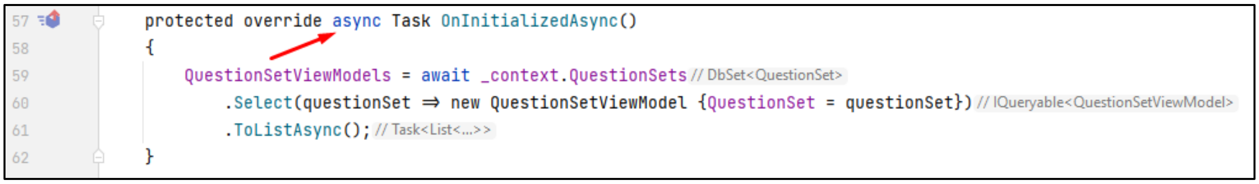


```
<Virtualize Items="Users">
    <ItemContent>
        <tr>
            <td class="align-middle">@context.Id</td>
            <td class="align-middle">@context.Email</td>
            <td class="align-middle">
                <button type="button" class="btn btn-danger" @onclick="() ⇒ DeleteUser(context)">Delete</button>
            </td>
        </tr>
    </ItemContent>
    <Placeholder>
        <tr>
            <td colspan="3">
                <div class="d-flex justify-content-center">
                  <div class="spinner-border" style="..." role="status">
                    <span class="sr-only">Loading...</span>
                  </div>
                </div>
            </td>
        </tr>
    </Placeholder>
</Virtualize>
```

Figure 13. <Virtualize /> example in "ManageParticipants.razor".

Asynchronous Programming

Asynchronous programming is applied while coding this application to prevent blocking the UI thread so that the UI is responsive to user’s input and feels snappy.



```
protected override async Task OnInitializedAsync()
{
    QuestionSetViewModels = await _context.QuestionSets // DbSet<QuestionSet>
        .Select(questionSet ⇒ new QuestionSetViewModel {QuestionSet = questionSet}) // IQueryable<QuestionSetViewModel>
        .ToListAsync(); // Task<List<...>>
}
```

Figure 14. Asynchronous programming example in "ManageQuestions/Index.razor".

Notes

- Please relog after seeding to avoid errors while answering questions.
- Every created user has default password of “d3F@uLtP@55w0Rd”.
- Default password can be changed in “appsettings.json”.
- Run seeding on every launch can be configured in “appsetting.json”.
- Used DBMS is SQL Server with database named “rdt” and uses Windows Authentication.

ERD

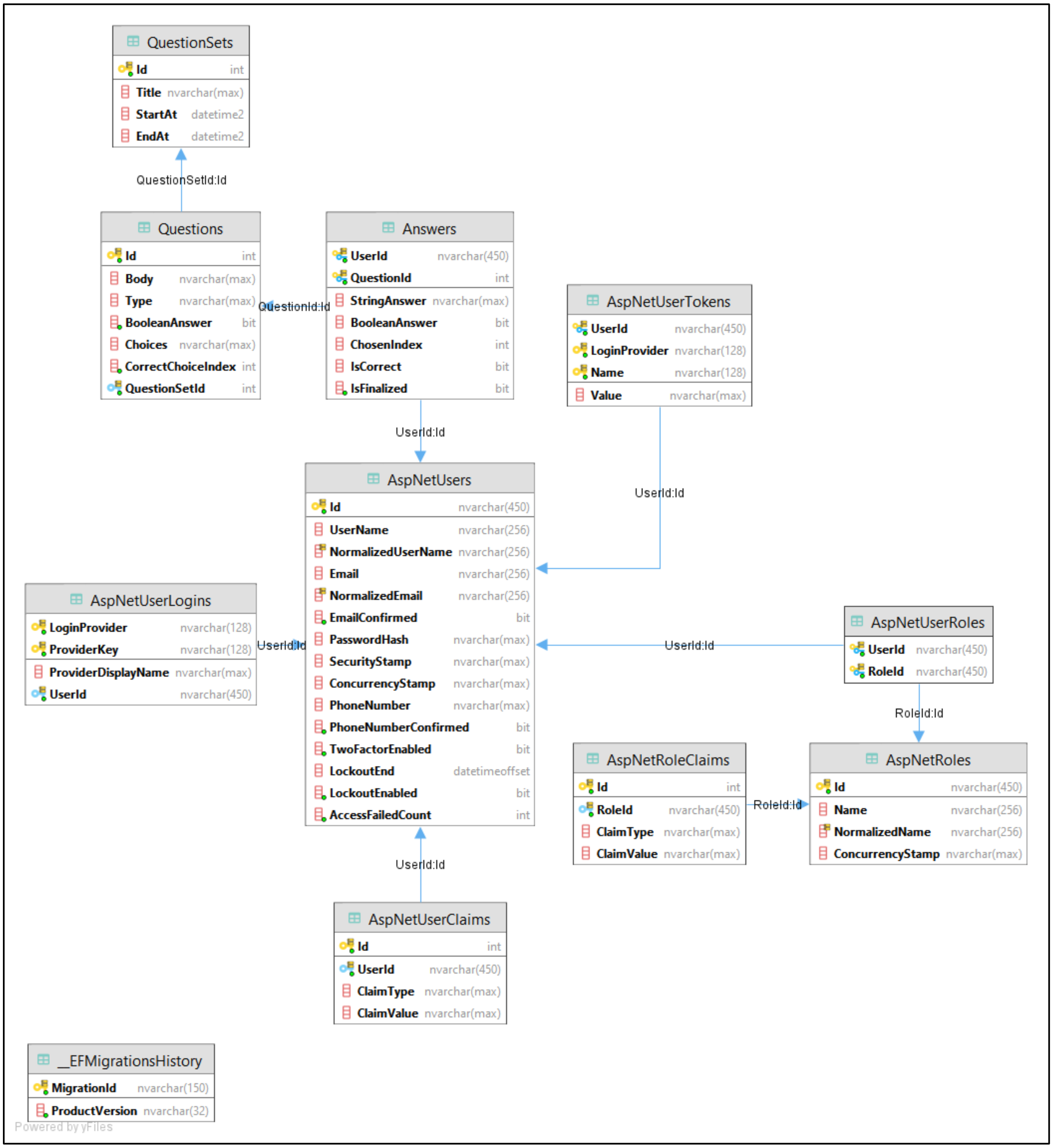


Figure 15. ERD

References

ASP.NET Web Apps | HTML, CSS, JavaScript, and C# (microsoft.com)

Choose between ASP.NET 4.x and ASP.NET Core | Microsoft Docs

Introduction to Identity on ASP.NET Core | Microsoft Docs

Blazor | Build client web apps with C# | .NET (microsoft.com)

ASP.NET Core Blazor hosting models | Microsoft Docs

Overview of Entity Framework Core - EF Core | Microsoft Docs

Introduction · Bootstrap (getbootstrap.com)

Table of Contents