International Society for Teacher Education (ISfTE)

Development Documentation

https://github.com/Extantdread/ISfTE
https://isfte.azurewebsites.net/

Admin Login Information:
Email: admin@admin.com
Password: Admin123$

Client: Dr. Ryan Cain and Dr. David Byrd
Spring 2020 Team: Robert Pickard, Michael Dansie, Derek Eaton

Table of Contents

Project Scope

Client Application

- Environment: .NET Core 3.1
- Language: C#, HTML, Javascript, JQuery
- Technology:
  - RazorPages
  - Entity Framework
- Packages:
  - Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore v3.1.2
  - Microsoft.AspNetCore.Identity.EntityFrameworkCore v3.1.2
  - Microsoft.AspNetCore.Identity.UI v3.1.2
  - Microsoft.AspNetCore.Mvc.Razor.RuntimeCompilation v3.1.2
  - Microsoft.EntityFrameworkCore.Sqlite v3.1.1
  - Microsoft.EntityFrameworkCore.SqlServer v3.1.2
  - Microsoft.EntityFrameworkCore.Tools v3.1.2
  - Microsoft.VisualStudio.Web.CodeGeneration.Design v3.1.1
- Static Details - ISfTE.Utility.SD
  - Prices, Roles, Descriptions, Client-Wide Values, Etc.
- Future Requirements:
  - Hosting Service with Email Capability
 - Business Paypal Account

User Interface

Initially, the development plan included another group which would design the UI for the site, but that fell through at the beginning of the project. Instead we implemented a simplistic, yet functional UI design through the use of Razor syntax, basic Html and JavaScript, and the following plugins:

- Bootstrap 4 Library[1]
- Font Awesome v5.10.2[2]
- SweetAlert JavaScript[3]
- DataTables JQuery[4]
- Toastr Alerts JavaScript[5]
- TinyMCE Text Editor[6]

Home Page (ALL USERS, Progress Bar for ATTENDEE)

Currently this page is somewhat of a placeholder. We do not have any graphics or information for the ISfTE conference, but they can easily be placed and formatted here as needed. When logged in as a conference attendee, a progress bar is displayed based on their user status which will indicate the next step they need to do in order to be completely registered for the conference.
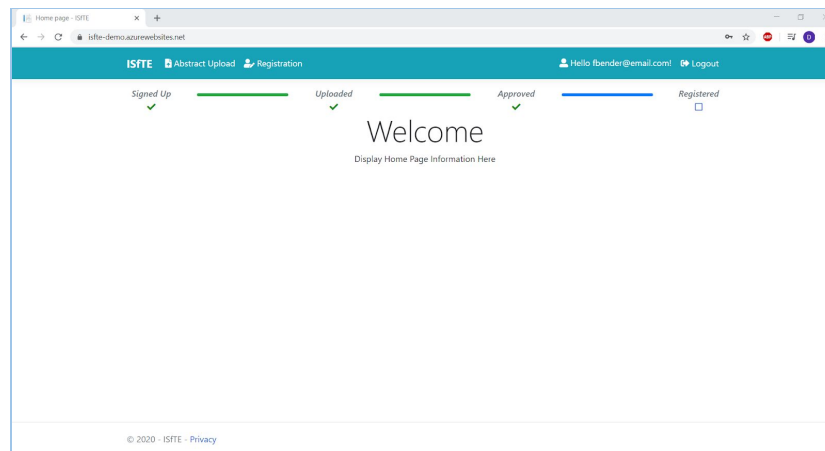
Abstract Rating Page (ADMIN and RATER)

The 'Index' page displays in a table all abstracts that have not been through a final review except for the current user's abstract, if it has been submitted. The table contains a link to view the abstract, some information about the abstract, the total number of current ratings, and the current user's rating, if it has been submitted.

Once a user clicks the 'Rate' or 'Review' button in the table, they will be taken to the rating 'Upsert' page. The user can view the abstract through this page as well as enter their ratings for the three given criteria through their slide-bar inputs and any comments through a TinyMCE input. The criteria descriptions can be pulled up in a SweetAlert pop-up by clicking on the criteria. The descriptions are stored in the SD class. If the user is 'reviewing' their previous ratings, all the criteria will be loaded to the current rating, and their previous comments will also be displayed.

Abstract Reviewal Page (ADMIN)

This 'Index' page is similar to the Abstract Rating 'Index' page. It will display ALL abstracts, except for the current user's, in a table with a link to the abstract, information about the abstract, the total number of ratings, the overall average rating, the status of the abstract, and a button that links to the 'Upsert' page for final reviewal.

The 'Upsert' page will also provide a link to view the abstract as well as each individual rating that has been submitted. The rating for each criteria, the average rating, and any comments will be displayed below the name, contact information, and date associated with the rating. The current user can then 'Approve' or 'Reject' the abstract, changing the status of the abstract and it's author depending on the selection.

Registration Page (ALL USERS)

This 'Upsert' page allows any user to register for the conference, after their Abstract has been approved. This page has many fields, including the optional sections 'Hotel Information' and '+1 Guest'. These sections are made using a Bootstrap accordion. This page is complex in using models. The ApplicationUser, Registration, Country, CountryType, and Guest models are all used on this page, and are connected through the RegistrationVM View Model. The list of Countries in the dropdown box is retrieved from the Database then alphabetically sorted using Javascript. There is a hidden CountryType dropdown box that is similar to the Country dropdown box. The purpose of this hidden box is to dynamically calculate the price, which is at the bottom of the page. The Country (and it's CountryType value of "Developed" or "Undeveloped"), and the presence of a Guest are used to calculate the price.

Admin Pages (ADMIN)

These pages are under a dropdown menu that is restricted to the Super Admin role. They are all DataTable based pages that allow admins to edit database values directly. The email button is a placeholder, no email functionality exists as of now.
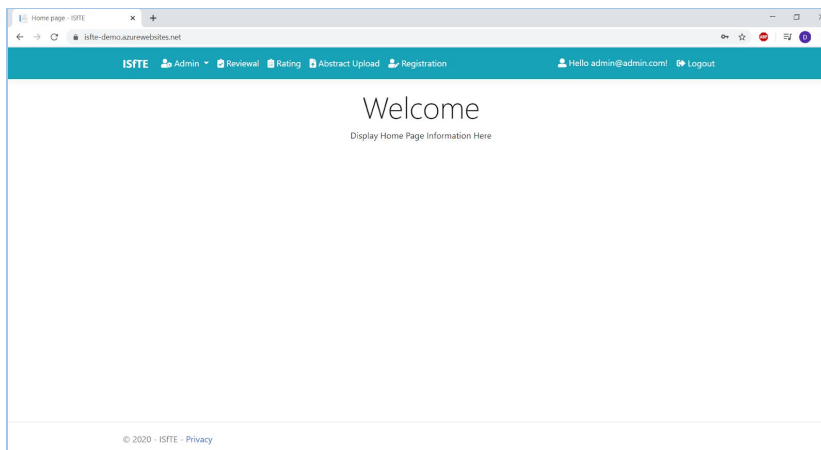
Users Page (ADMIN)

Display the list of users which can be filtered by searching for name, email, or status, and sorted by any of the displayed values. An admin is able to create a new Admin or Rater user, and is able to view a list of all users and their role or status. By clicking the 'Lock' button, an account can be suspended until 'Unlocked' by the same button in case of malicious users.
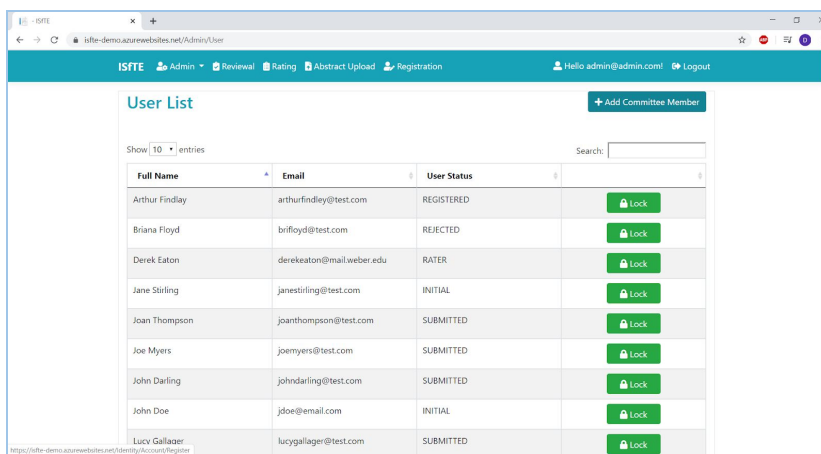
Registrations Page (ADMIN)

These pages are similar to the regular Registration pages, but they allow an admin to view and edit a different user's registration. The 'Index' page displays all registrations that have been submitted. The table contains a link to view the registration, and some information about the registration.

The 'Upsert' page is virtually the same as the Registration Page above, but it has additional backend code that allows an admin to edit the registration without editing their own.

Countries Page (ADMIN)

This page displays in a table all the countries currently in the database along with their country type (Developed or Undeveloped) for registration pricing. Countries can be edited, deleted, or created through the buttons on this page. We pulled the list of countries and classifications currently in the database from this 2019 report.[7]

Abstract Upload (ALL USERS)
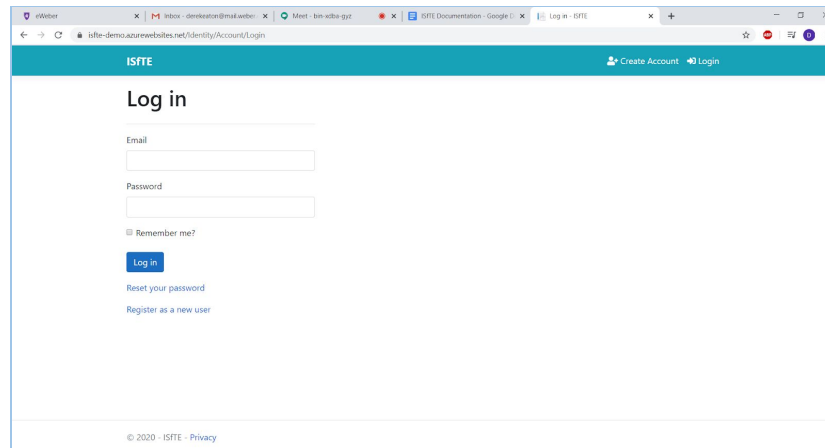
This page allows users to upload their abstract to be reviewed. It has a title field, and a file field which lets users select a file from their computer. There are security features that do not allow a file greater than 1MB to be submitted, and the file's extension type must be a .PDF, .DOC, or .DOCX file.

Create Account (ALL USERS)

This page is based off the default Create Account page provided by .NET Core, but has some additions. It uses the ApplicationUser model, and has more fields, such as Institution, Title, and Preferred Name. It tells the user which fields are required.

Login (ALL USERS)
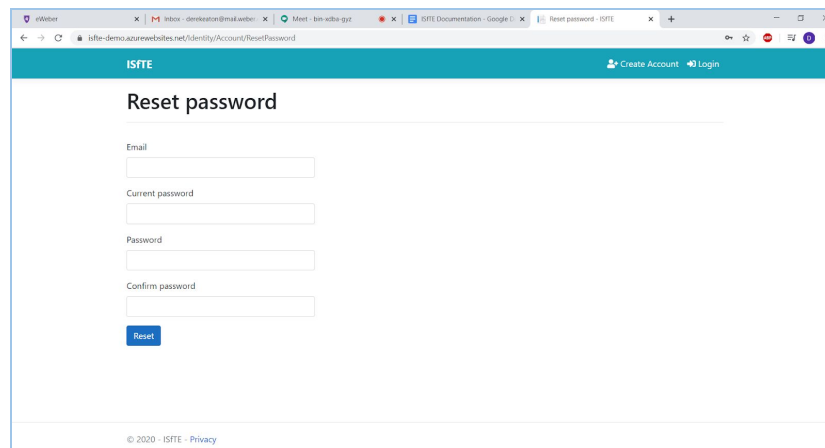
This page uses the default Login page provided by .NET Core, it requires an email and password to login. It also has links to the Create Account and Reset Password pages.

Reset Password (ALL USERS)

This page allows a user to change their password. The user must know their email and old password to do this. The only link to this page is on the Login page.

Hosting

We are currently hosting the testing site on Azure. To do this, you need an Azure for Students account. The host can be changed at any time by the request of the client, if a hosting service is provided. The final server would need to support email.

Database

Database Overview

We used Weber State's Titan SQL Server. Future groups will most likely have to recreate the database. We used the Titan SQL Server because it was already set up, and it allowed all group members to access the same data. Connection info for this server will most likely be changed in future semesters. You can find the connection info in the appsettings.json file. We built the database through Visual Studio, using Entity Framework. You should be able to build the database easily using Entity Framework by following these steps.

We have included SQL data files in the repository under the SQL Data Population folder to populate the Country and CountryType tables, using the 2019 Report mentioned earlier.

How To Build the Database:

1. Pull the repository.
2. Install correct Nuget Packages listed above.
3. Enter new server connection information for Titan (or other DB) under appsettings.json.
4. In Package Manager Console, create an initial migration using the command "add-migration MigrationName"
5. In Package Manager Console, use the command "update-database".

ERD DIAGRAM

ApplicationUser(Inherits from Identity)
- FirstName (string)
- LastName (string)
- Institution (string)
- Title (string)
- Preferred Name (string)
- Status (string)

Abstract

PK	Id (int)
FK	AuthorId (string)
	DocPath (string)
	Title (string)
	Approved (boolean)
	ReviewDate (DateTime?)

AbstractRating

PK	Id (int)
FK	AbstractId (int)
FK	RaterId (string)
	Mechanics (int)
Key	ConferenceFit (int)
Key	Contribution (int)
	Comments (string)

Registration

PK	Id (int)
FK	AttendeeId (string)
	Organization (string)
	StreetLine1 (string)
	StreetLine2 (string)
	TerritoryState (string)
	PostalCode (string)
FK	CountryId (int)
	Fax (string)
	PreferredName (string)
	ArrivalTransport (string)
	ArrivalDateTime (DateTime?)
	DepartTransport (string)
	DepartDateTime (DateTime?)
	DietDisabilityNeeds (string)
	HotelInDate (DateTime?)
	HotelOutDate (DateTime?)
	Hotel (string)
	TotalCost (double)

Guest

PK	Id (int)
FK	RegistrationId (int)
	FirstName (string)
	LastName (string)
	Email (string)
	Relationship (string)

Country

PK	Id (int)
	Name (string)
FK	CountryTypeId (int)

CountryType

PK	Id (int)
	Description (string)

Research, Tutorials, and Resources

Visual Studio NuGet Packages

In Visual Studio, go to Tools → NuGet Package Manager → Manage NuGet Packages for Solution.

Entity Framework

Links Included in the Document

[1] https://getbootstrap.com/docs/4.4/getting-started/introduction/
[2] https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use
[3] https://sweetalert.js.org/guides/
[4] https://datatables.net/
[5] https://cdnjs.com/libraries/toastr.js/latest
[6] https://www.tiny.cloud/
[7] http://hdr.undp.org/sites/default/files/hdr_2019_overview_-_english.pdf
[8] https://azure.microsoft.com/en-us/free/students/starter/
[9] https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/intro?view=aspnetcore-3.1&tabs=visual-studio

Future Plans and Features

There are several features that can be added at a later date. This section will cover some of the planned features that will be implemented.

Territory State Dropdown

The implementation of an extension to the country selection tools. Namely, the ability to specify a Territory or State to more accurately state the place of residence. It was suggested by Dr. Fry that this should be implemented through an API. This would populate the Territory State field of the Registration page.

Print Name Badges

Implement a way to create name tag badges for the people whose abstracts have been accepted and have registered for the conference. These will need to be handled so that when participants arrive, they can receive their name tag. This could be included through the 'User' page of the Admin dropdown.

Email

Add an auto-email function when an abstract is approved or denied so that the person that made the submission is notified automatically. This could be included as another part of the OnPost method of the 'Upsert' Abstract Reviewal page. Admins also have an email link in the dropdown which is currently not functional. Here they could send out emails to groups of users to easily communicate information about the conference as well as provide updates or instructions for raters and other admins. This feature could also be used to reset passwords or verify emails on sign-up. The current method to reset your password requires you to know your current password.

Paypal

Add a way to pay through Paypal when a user is registering. To fully implement a Paypal account, you will need the login details for a Business Paypal account.