

Overview of the Organization

Globally, **SD Worx** offers a broad range of products to its clientele, including HR and payroll management, legal assistance, training, automation, consulting, and outsourcing. Currently, more than 63,000 big and small businesses worldwide rely on SD Worx's more than 70 years of experience.

Almost 3,900 people work for SD Worx across 10 nations, which include the United Kingdom, Austria, France, Germany, Ireland, Luxembourg, Mauritius, the Netherlands, Switzerland, and Belgium, where the company is headquartered.

Some insight about SD Worx, between 1945-1960, employers in Belgium create SD Worx, a wage administration company. Then around 1960, SD Worx became the first payroll company to embrace technology. It continues to grow through innovation, extending its offering beyond payroll. Around 2000, SD Worx starts its European journey and expands into Luxembourg, the Netherlands, France and Germany. Around the 2010s, SD Worx introduces the first pan-European payroll solution. In 2015, SD Worx Global Payroll Network coverage increases to include the US and Canada. Then finally the company touches our island, Mauritius in 2016. SD Worx is present in Benelux, France, UK, Ireland, Germany, Austria, Switzerland and Mauritius. Payroll Services Alliance coverage includes Asia. SD Worx acquires Protim, market leader in Workforce Management. SD Worx expands into the flexible labour market, with the acquisition of Vio and Flexpoint.

Particularly well-known for its proficiency in payroll services is SD Worx. This entails overseeing intricate payroll procedures, making certain that regional laws are followed, and providing solutions for businesses of various sizes. SD Worx uses creativity and technology to deliver effective and user-friendly solutions. They might provide software platforms and solutions with features like automation, analytics, and self-service choices to make payroll and HR management easier.

And not to forget to mention its security, SD Worx places a high priority on data security and compliance with local laws because HR and payroll data is sensitive. Ensuring the secure handling of client data in compliance with privacy rules is crucial.

2.1 Customer Experience

1. Commitment drives us forward

We have the will to succeed and we feel the need to go forward. We take pride and ownership in our work, from beginning to completion. Everyday, we take charge of the situation and look after our customers by guiding them in the right direction. No matter how big the challenge or how many stumbling blocks along the way, we are dedicated to deliver results. With confidence, we look ahead and keep on moving, finding the way to new solutions.

2. Dear Customer

We always put our best foot forward for our customers. By placing their needs and requirements in front of anything and everything else. We keep them ahead of the game. Yet, we manage expectations and are not afraid to challenge questions. We really listen and put ourselves in the shoes of the people that use our products and services. By truly understanding their goals, we use our expertise to provide the best quality service with passion and professionalism. Placing the customers first is a team achievement: we are all responsible and we all have a role to play.

3. One for all, all for one

We are close and feel interconnected, even if we're physically apart. With the interest of the team at heart, we all pull together with a clear view on where we're going. All team members are equal partners, and everyone knows their roles & responsibilities. Even though we share our differences, we always keep the dialogue going. With respect for different opinions, we understand that we can achieve so much more when we join forces. Working closely together triggers a sense of belonging and camaraderie: we're all part of the SD Worx family.

4. The road is open

We dare to look at things from a different perspective. Not always knowing what comes next, we look in all directions to form new ideas to make things better. Innovation is in our DNA and imagination is our best friend. It inspires creativity and pushes us to see possibilities beyond the realities and pushes us to see possibilities beyond the realities of today. Looking into the future, we accept new challenges without prejudice. We are open to learn new things and to question old habits. This requires us to be bold and positive. This requires us to be bold and positive. We can only be truly open minded if we rely on our spontaneous enthusiasm, without taking life all too seriously.

5. We believe in each other

Trust and integrity are what we stand for. We have faith in each other to do what's expected. Both our colleagues and our customers rely on us to look after them and take them in the right direction. We have confidence in our own abilities and feel empowered to make our own decisions, knowing that the team will always support us. Trust and honesty are closely tied together, we're sincere and we dare to set boundaries. In other words, we stick to our promises but we only promise what we can keep.

3.0 Technical Content

During these 10 weeks, I actively engaged in a range of projects, demonstrating my commitment to advancing my skills as a Full Stack Developer and also in Quality Assurance Testing. Over this period, I had the opportunity to contribute significantly, to a project called 'SDEvents' where I was able to implement some key tasks. Additionally, I was trained for quality assurance testing where I tested multiple scenarios of an online task to see if any bugs arise. The fast-paced and dynamic environment at [Company Name] provided an ideal platform for me to apply theoretical knowledge gained during my academic studies in a practical setting. Successfully passing this internship has not only bolstered my technical proficiency but has also instilled in me a deeper appreciation for collaboration, problem-solving, and the intricacies of real-world software development.

3.1 Training

My internship training was divided into two stages

1. Full Stack Developer
2. Quality Assurance Testing

1. Full Stack Developer

I engaged in comprehensive training by utilizing resources such as [W3Schools](#), [YouTube](#), [Angular](#) and [GitHub](#). We were given the task to read and watch tutorial on W3Schools to learn C#. The tutorials on w3schools.com provided a structured and informative foundation for both front-end and back-end development, covering essential languages like HTML, CSS, JavaScript, and server-side scripting languages such as Node.js. The platform's hands-on examples and interactive exercises were instrumental in reinforcing theoretical concepts and honing practical skills.

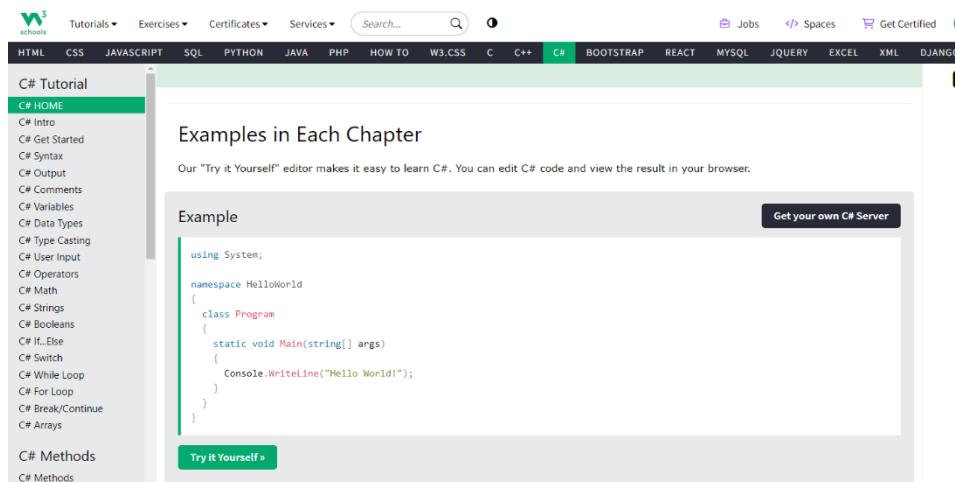
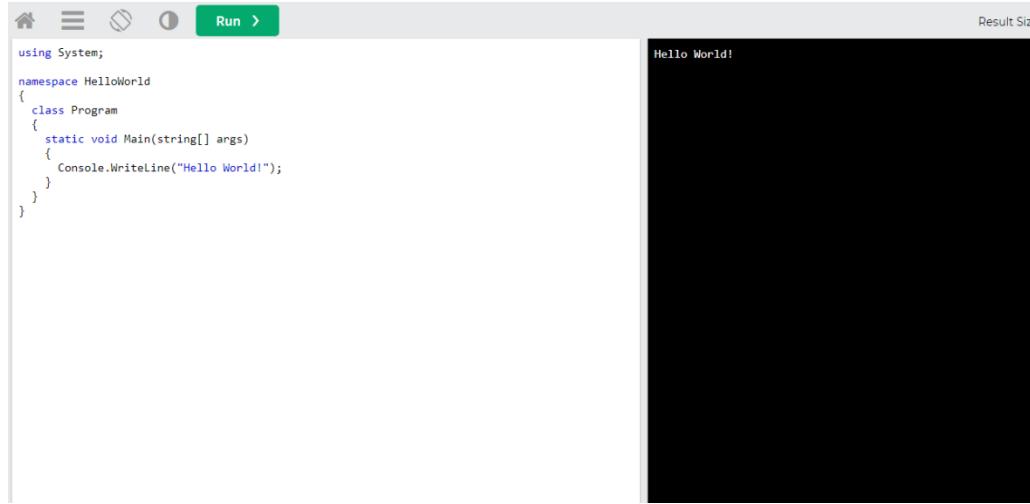


Figure 1 shows W3School Website



The screenshot shows the W3School Online Code Editor interface. On the left, there is a code editor window with the following C# code:

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

On the right, there is a results window titled "Result Size" which displays the output "Hello World!".

Figure 2 shows W3School Online Code Editor

Learning C# has been a rewarding experience as it introduced me to a powerful and versatile programming language with a wide range of functions. As I delved into C#, I gained an understanding of its various functions, including its robust type system, support for polymorphism, and extensive standard libraries. C# is particularly notable for its role in developing Windows applications, web services through ASP.NET, and game development with Unity. Its LINQ (Language-Integrated Query) capabilities facilitate seamless data querying and manipulation, while asynchronous programming features enhance performance in handling concurrent tasks. The integration of C# with the .NET framework further expands its capabilities, providing a user-friendly shell for application development. Overall, learning C# has equipped me with a versatile skill set, enabling me to tackle a diverse array of programming challenges.

3.2 Application Used

1. Visual Studio 2019 & 2023

Visual Studio, an IDE made by Microsoft, provides developers a feature rich development environment to develop high-quality code efficiently and collaboratively. We can write, edit, debug, build, and deploy our application. Compilers, code completion tools, source control, extensions, and many more capabilities are included in Visual Studio in addition to code editing and debugging to improve each step of the software development process. With Visual Studio's wide range of capabilities and language support, you may write the simplest program of all to creating and implementing apps. For instance, create, debug, and test applications in C++ and .NET, edit ASP.NET pages in the web designer view, use .NET to create cross-platform desktop and mobile applications, or use C# to create responsive Web user interfaces.

The code editor in Visual Studio offered advanced functionalities such as syntax highlighting, auto-completion, and code navigation, significantly enhancing my coding speed and accuracy. The built-in debugging tools proved invaluable, enabling me to identify and resolve issues swiftly during the development process.

For SDWorx, both Visual Studio 2019 and 2023 were used for backend and frontend purposes. In Visual Studio 2019, we primarily used C# for the Application Programming Interface(API) and in the 2023 one, we used TypeScript, HTML, CSS for the User Interface.

Visual Studio's integration with various version control systems, including Git, streamlined collaborative development, ensuring smooth collaboration with team members. The IDE's support for a wide range of programming languages, frameworks, and technologies made it versatile and adaptable to the diverse requirements of my projects. In addition to its coding capabilities, Visual Studio provided an integrated testing environment that facilitated the Quality Assurance phase. The seamless integration of testing frameworks and debugging tools within the IDE contributed to a comprehensive and efficient testing process.

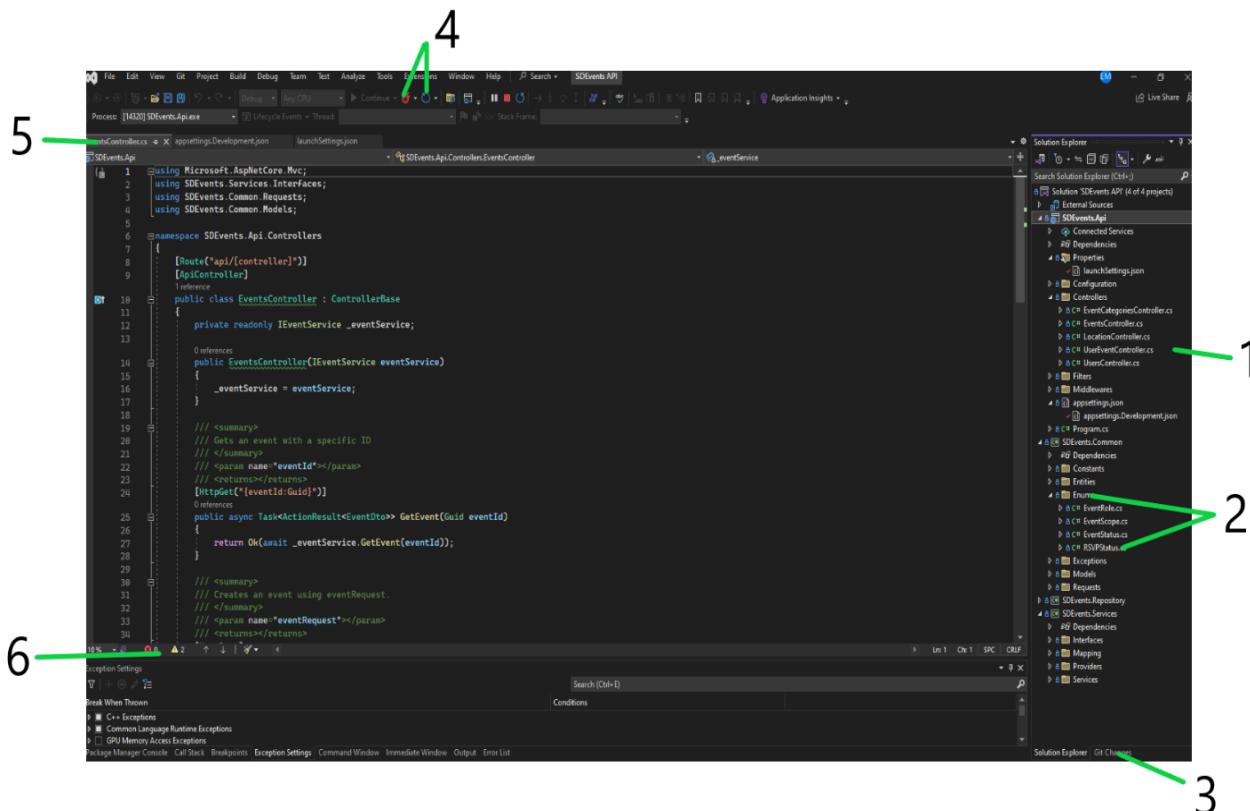


Figure 1 shows Visual Studio 2019 IDE

1. Solution Explorer to view files and folders, such as classes or controllers.
2. Folders and C# files
3. Git Changes, to collaborate on code projects with the team
4. Run or stop code
5. Tabs to navigate between opened codes
6. Panel which shows error and warning messages

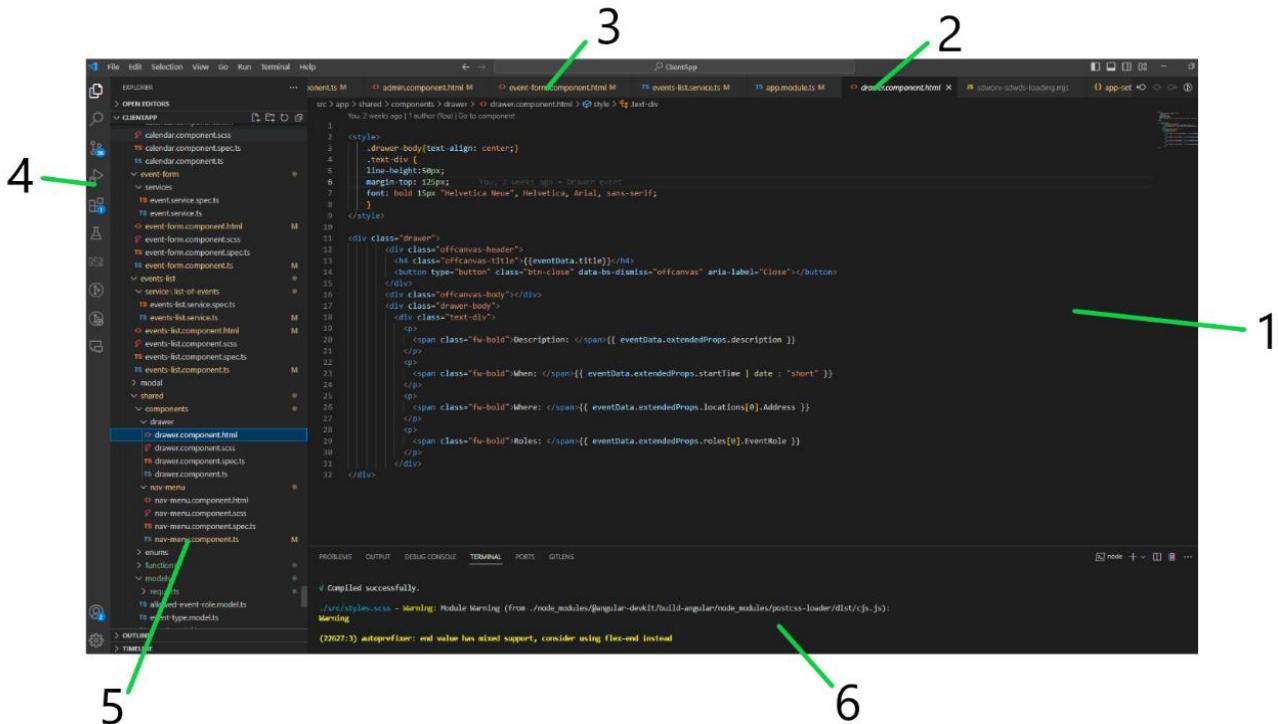


Figure 2 shows Visual Studio 2023 IDE

1. Code Section
2. Opened tab which displays its code in the code section
3. Other opened tabs
4. Explorer section such as files, search, source control, extensions & settings
5. Folders and files section, Typescript, JavaScript, CSS and HTML
6. Terminal and output window to run codes

2. Microsoft SQL Server Management Studio

An integrated platform for administering any SQL infrastructure, from SQL Server to Azure SQL Database, is SQL Server Management Studio (SSMS). Tools for configuring, monitoring, and managing SQL Server and database instances are offered by SSMS. Build queries and scripts using SSMS and use it to deploy, monitor, and upgrade the data-tier components that your applications rely on.

It was used mostly in my task to:

1. To connect to as Database server
2. To insert, delete and update Tables
3. To test input and output when utilizing the application.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "MININT-1OQA07J.sdevents - dbo.Users - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Project, Query Designer, Tools, Window, Help. The toolbar has various icons for connecting, disconnecting, executing queries, and managing objects. The Object Explorer on the left shows the database structure, including the "sdevents" database and its tables like "Tables", "System Tables", "FileTables", "External Tables", "Graph Tables", and "dbo.Users". The "dbo.Users" table is selected and displayed in the main grid. The table has columns: UserId, RoleId, Email, Name, and IsActive. The data grid shows 10 rows of user information. At the bottom, there are navigation buttons for the table.

UserId	RoleId	Email	Name	IsActive
ted35d7az2	ceebabd99-6825-4...	kalim.auckbaraullee@com...	Kalim Auckbaraullee	False
2fd8bb6...	ceebabd99-6825-4...	ridhwaan.paraouty@com...	Ridhwaan Paraouty	False
1a16b94...	e244aca56-91df-4...	esiae.moos@company.co...	Esaie Moos	False
ac70f2c7...	ceebabd99-6825-4...	yusra.cader@company.co...	Yusra Cader	False
54a9bb6...	ceebabd99-6825-4...	dharun.teemul@company...	Dharun Teemul	False
f4f0cd3...	e244aca56-91df-4...	shivani.ramalingum@com...	Shivani Ramalingum	False
c0a803f1...	ceebabd99-6825-4...	abhisarika.jeebodhun@co...	Abhisarika Jeebodhun	False
ba1f59f4...	ceebabd99-6825-4...	manav.rammarain@comp...	Manav Rammarain	False
32d04b5...	ceebabd99-6825-4...	doorgesh.neetye@compa...	Doorgesh Neetye	False
4a46bbd...	e244aca56-91df-4...	khilesh.chumbit@compan...	Khilesh Chumbit	False
*	NULL	NULL	NULL	NULL

Figure 3 shows Microsoft SQL Server with data

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects under the schema 'dbo'. In the center, there are three tabs: 'MININT-1OQA07J.sdevents - dbo.Events', 'MININT-1OQA07J.s...o.EventCategories', and 'MININT-1OQA07J.sdevents - dbo.Roles'. The first tab is selected and shows a table with the following data:

EventId	Location...	Name	StartTime	EndTime	Description	Status	Scope	Categor...
198784a09d...	8650e11...	Photography Session	2023-11-12 00:00:00.000	2023-11-14 00:00:00.000	Employee photo session	1	2	Hackathon
070fd86...	b8a568b...	Ti Alexandre Concert	2023-11-27 00:00:00.000	2023-11-29 00:00:00.000	Ti Alexandre Music Festival2	0	3	Hackathon
00a64e5...	8b17b8c...	EoY Party	2023-11-09 00:00:00.000	2023-11-10 00:00:00.000	End of Year Party	2	1	Hackathon
6bc46ecf...	d9bde39...	Khilesh Party	2023-11-25 00:00:00.000	2023-11-26 00:00:00.000	Khilesh Party	2	1	Hackathon
3ad9aa7...	e8a74c7a...	Christmas Party	2023-11-24 00:00:00.000	2023-11-24 00:00:00.000	Christmas celebration in the office	1	2	Hackathon
92b7bb2...	8b17b8c...	SD Pool Tournament	2023-11-25 00:00:00.000	2023-11-30 00:00:00.000	The SD Pool Tournament is an ann...	0	0	Hackathon
53dbdd9...	9e34d67...	Pizza Party	2023-11-23 00:00:00.000	2023-11-24 00:00:00.000	Ridwaan paying pizza	1	2	Hackathon
f018529...	8650e11...	Yogesh Party	2023-11-25 00:00:00.000	2023-11-26 00:00:00.000	Yogesh Party	2	1	Hackathon
c18b538...	d9bde39...	Sparkathon	2023-11-02 00:00:00.000	2023-11-04 00:00:00.000	3-day hackathon	1	1	Hackathon
5e37fbe0...	b8a568b...	Food Festival	2023-11-17 00:00:00.000	2023-11-19 00:00:00.000	Sale of various food items	2	1	Hackathon
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4 shows Microsoft SQL Server with data

3. Angular

Tour of Heroes App

Angular is a development platform and application design framework for building smart and effective single-page applications. At its core, Angular employs TypeScript, a superset of JavaScript, which enhances the maintainability and scalability of code through static typing. The framework follows the Model-View-Controller (MVC) architecture, promoting a modular and organized structure for developing complex applications. Angular's two-way data binding ensures seamless synchronization between the model and view, reducing boilerplate code and enhancing developer productivity. Its comprehensive set of directives, such as **ngFor** and **ngIf**, simplifies DOM manipulation, and the dependency injection system facilitates modular and testable code. Angular's ability to create reusable components and its strong support for creating dynamic and responsive user interfaces make it a preferred choice for building modern web applications. Additionally, Angular integrates smoothly with back-end technologies, allowing for efficient communication with server-side components. Overall, my exploration of Angular has provided me with a powerful toolset for crafting interactive and scalable user interfaces in conjunction with my full-stack development skills.

After grasping the Angular knowledge, we were tasked to work on our first Angular project which was called 'Tour of Heroes'.

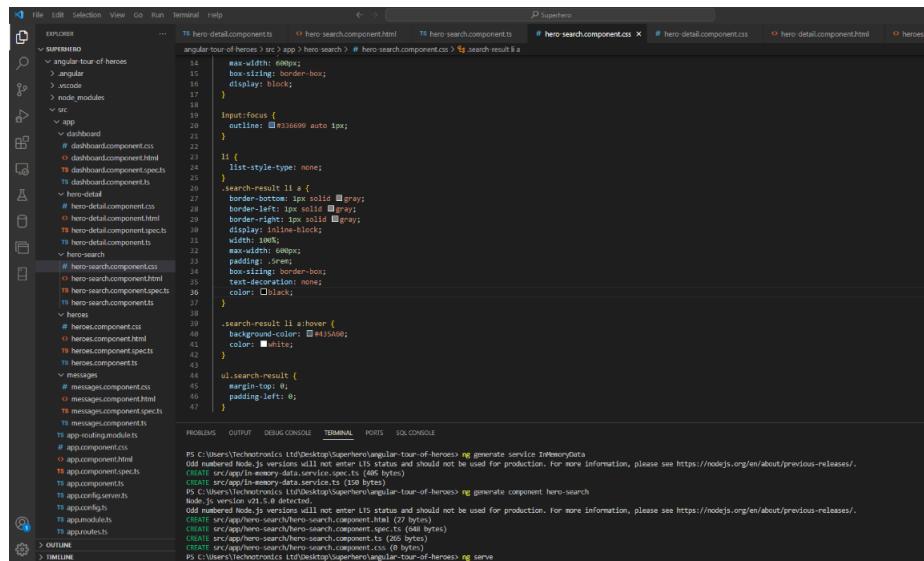
The Tour of Heroes application that you build helps a staffing agency manage its stable of heroes. The application has many of the features that you'd expect to find in any data-driven application.

The finished application:

- Gets a list of heroes
- Displays the heroes in a list
- Edits a selected hero's details
- Navigates between different views of heroic data

This tutorial helped in gain confidence that Angular can do whatever you need it to do by showing how to:

- Use Angular directives to show and hide elements and display lists of hero data.
- Create Angular components to display hero details and show an array of heroes.
- Use one-way data binding for read-only data.
- Add editable fields to update a model with two-way data binding.
- Bind component methods to user events, like keystrokes and clicks.
- Enable users to select a hero from a list and edit that hero in the details view.
- Format data with pipes.
- Create a shared service to assemble the heroes.
- Use routing to navigate among different views and their components.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "angular-tour-of-heroes".
- Code Editor:** Displays a CSS file named "hero-search.component.css". The code includes styles for a search input and a list of search results.
- Terminal:** Shows the command-line output of running the application. It includes Node.js version information, service generation commands, and a "serve" command.
- Bottom Status Bar:** Shows tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", "PORTS", and "SQL CONSOLE".

```

14   max-width: 600px;
15   box-sizing: border-box;
16   display: block;
17 }
18
19 input:focus {
20   outline: #e33669 auto 1px;
21 }
22
23 li {
24   list-style-type: none;
25 }
26 .search-result li & {
27   border: 1px solid #gray;
28   border-left: 3px solid #gray;
29   border-right: 3px solid #gray;
30   display: inline-block;
31   padding: 5px;
32   width: 600px;
33   padding: 5px;
34   box-sizing: border-box;
35   text-decoration: none;
36   color: black;
37 }
38
39 .search-result li:hover {
40   background-color: #e3e3e3;
41   color: white;
42 }
43
44 ul.search-result {
45   margin-top: 0;
46   padding-left: 0;
47 }

```

```

PS C:\Users\Technopedia\Documents\Visual Studio Code\angular-tour-of-heroes> ng generate service InMemoryData
Node.js:13:23: warning: 'use strict'; is not needed in a module and should not be used for production. For more information, please see https://nodejs.org/en/about/previous-releases/.
CREATE src/app/in-memory-data.service.ts (140 bytes)
CREATE src/app/in-memory-data.service.spec.ts (48 bytes)
PS C:\Users\Technopedia\Documents\Visual Studio Code\angular-tour-of-heroes> ng generate component hero-search
Node.js: version v11.5.0 detected
Node.js: numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see https://nodejs.org/en/about/previous-releases/.
CREATE src/app/hero-search/hero-search.component.ts (26 bytes)
CREATE src/app/hero-search/hero-search.component.html (27 bytes)
CREATE src/app/hero-search/hero-search.component.spec.ts (48 bytes)
CREATE src/app/hero-search/hero-search.module.ts (26 bytes)
PS C:\Users\Technopedia\Documents\Visual Studio Code\angular-tour-of-heroes> ng serve

```

Figure 3 shows VS Code for Tour of Heroes

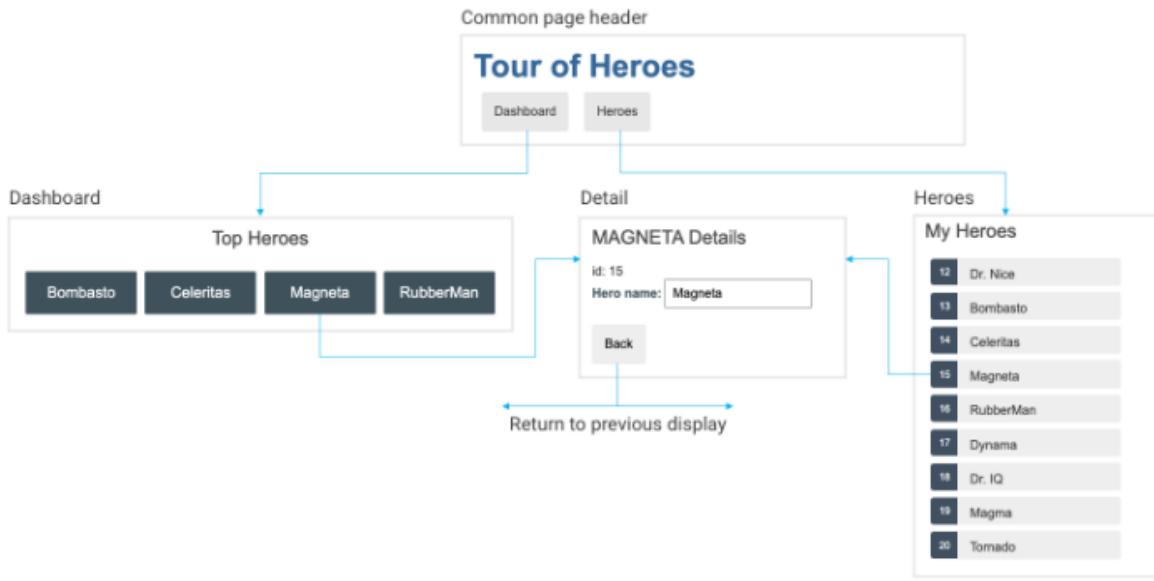


Figure 4 shows breakdown of Tour of Heroes App

CRUD with Angular UI and API

I was given the opportunity to learn CRUD(Create, Read, Update, Delete) application using Angular 14, .NET 6, and SQL Server. We were guided by our mentors and also used a detailed explanation tutorial of a YouTube video by Patrick God, who gives in depth explanation about coding in general.

The tutorial begins by creating an Angular 14 application, defining a superhero model, and generating a service for handling communication with the backend API. On the backend, a .NET 6 Web API project is established, including the definition of a superhero model, the creation of a corresponding controller, and the implementation of CRUD operations.

The Angular application is then configured to make HTTP calls to the .NET API. CORS is enabled, and Entity Framework Core 6 is integrated to interact with a SQL Server database. The tutorial covers aspects such as setting up a connection string, installing EF Core tools, registering the `DataContext`, and running initial migrations. Detailed steps for retrieving superhero data from the database, implementing CRUD operations in the API, and creating an Angular component for editing superhero details are provided.

The tutorial concludes with final tests to ensure the seamless functionality of all CRUD operations, offering a comprehensive guide to building a robust full-stack application.

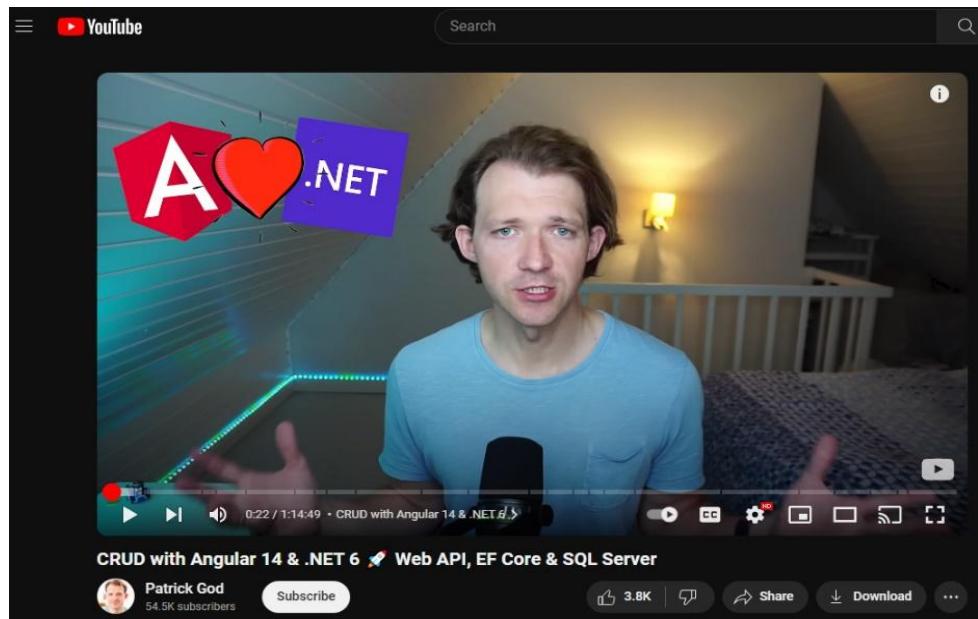


Figure 5 shows the YouTube tutorial video

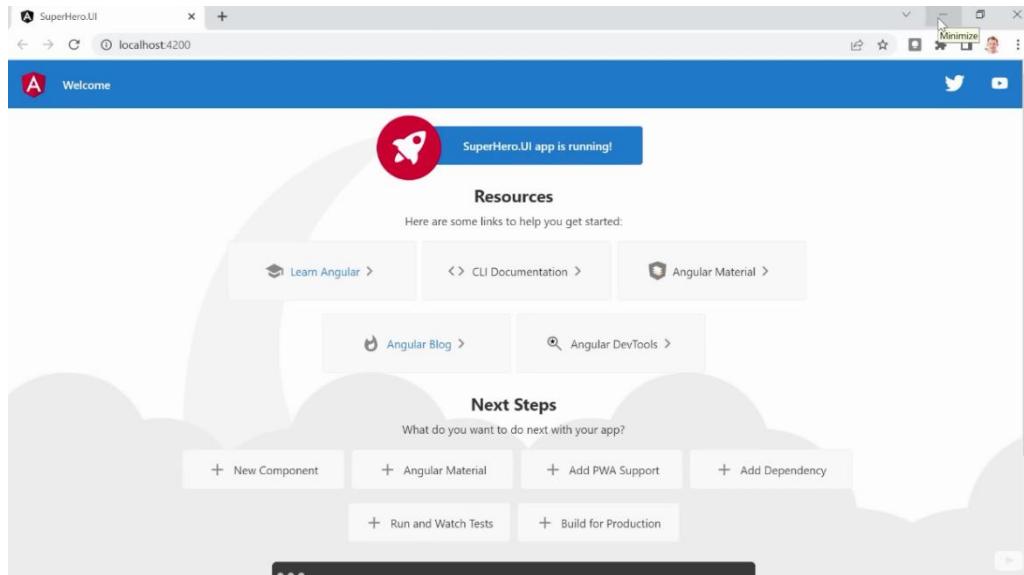


Figure 6 shows Angular scratch homepage

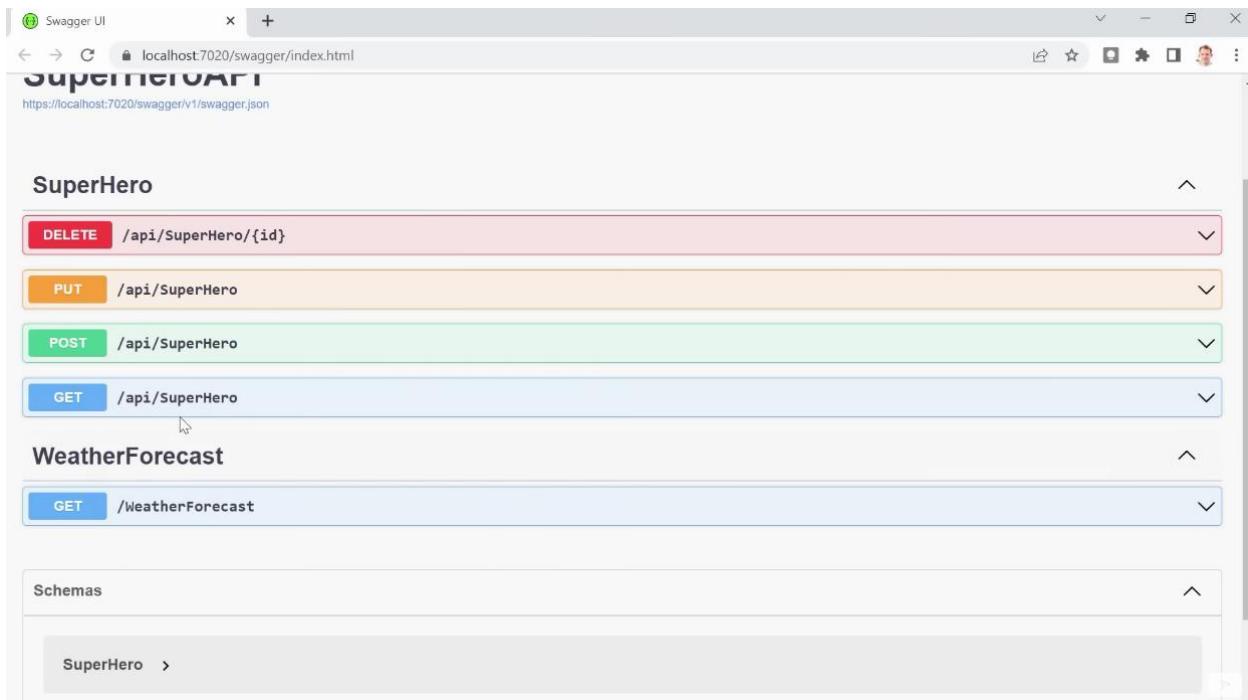


Figure 7 shows the SuperHero API swagger

With the help of Visual Studio 2019, I was able to make the SuperHero API. I started by initiating a new project and selecting the ASP.NET Core Web API template, leveraging the power of .NET 6 for backend development. Within Visual Studio, I meticulously defined the SuperHero model, incorporating essential attributes and data types. To streamline API documentation and testing, I utilized Swagger, which describes the structure of the API, by integrating the **Swashbuckle.AspNetCore** package.

I was able to generate an interactive API documentation interface, facilitating a smooth exploration of available endpoints. The API's CRUD operations were implemented in the SuperHeroController, handling HTTP requests for creating, reading, updating, and deleting superhero data. Visual Studio's code navigation and debugging tools were instrumental in ensuring the seamless integration of Entity Framework Core 6, establishing a connection to a SQL Server database.

The migration process, orchestrated with Entity Framework Core tools, efficiently translated the model into a relational database schema. Overall, Visual Studio 2019 provided a robust and user-friendly environment for developing, testing, and documenting the SuperHero API with Swagger, making the entire CRUD process intuitive and efficient.

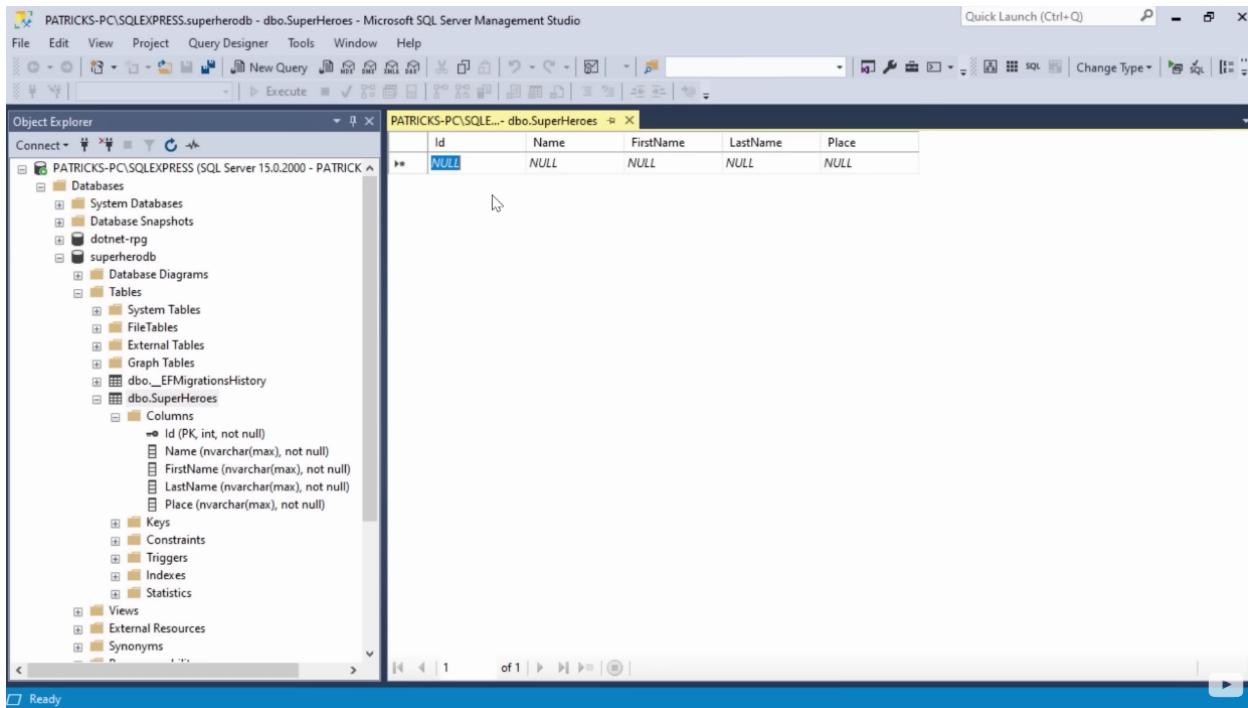


Figure 8 shows SQL Server with SuperHeroes Database

In conjunction with Visual Studio 2019, Microsoft SQL Server played a pivotal role in the development of the SuperHero API. Leveraging the Entity Framework Core within the .NET 6 framework, I seamlessly integrated SQL Server as the backend database for storing and managing superhero data.

Within Visual Studio, I configured the connection string to establish a secure link between the API and the SQL Server database, ensuring smooth communication. The database schema was designed to align with the SuperHero model, and with the assistance of Entity Framework Core's Code-First Migrations, I executed the initial migration to create the necessary tables.

This approach not only facilitated the storage of superhero information but also streamlined the process of implementing CRUD operations. Microsoft SQL Server's reliability, scalability, and seamless integration with Visual Studio provided a robust foundation for building a resilient backend infrastructure for the SuperHero API, enhancing data management and persistence capabilities.

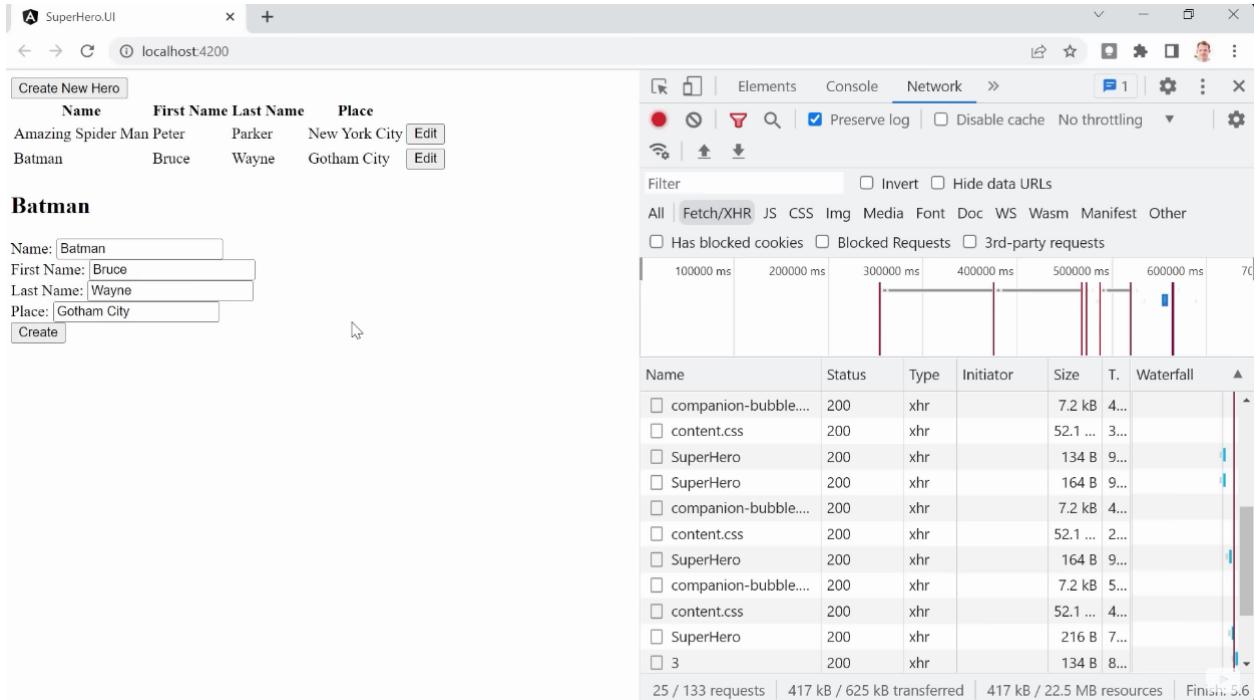


Figure 9 shows SuperHero UI

As I developed the SuperHero UI using Angular 14, I employed various tools and features in my browser, such as the browser's Developer Tools, to inspect and analyze the application's elements and network interactions.

With the help of the Elements tab, I meticulously examined the DOM structure, scrutinizing the HTML and CSS elements rendered by the Angular components. This enabled me to identify and resolve any styling or layout issues, ensuring a polished and cohesive user interface.

Additionally, the Network tab provided valuable insights into the data-fetching process. I closely monitored the HTTP requests and responses exchanged between the Angular client and the .NET 6 Web API.

By examining the network activity, I gained a deeper understanding of how data was being retrieved from the backend, helping me optimize performance and troubleshoot potential issues. This hands-on approach to inspecting elements and monitoring network activity proved instrumental in fine-tuning the SuperHero UI, ensuring a seamless and responsive user experience.

4. GitHub

GitHub is a widely-used web-based platform that serves as a collaborative environment for software developers to host, share, and collaborate on code repositories. Offering version control through Git, GitHub facilitates efficient collaboration among teams, enabling developers to work on projects simultaneously while maintaining a detailed history of changes.

Its intuitive interface allows users to create repositories, manage branches, and propose changes through pull requests. GitHub is not only a hosting service but also a social platform, fostering community engagement through features like issue tracking, discussions, and wikis. It has become an integral part of the software development lifecycle, providing a centralized hub for open-source contributions and enabling developers worldwide to collaborate on diverse projects seamlessly.

The platform's robust set of features, including actions for continuous integration, code review tools, and integration with various third-party services, makes GitHub a cornerstone in modern software development workflows.

5. Azure DevOps

Azure DevOps is a comprehensive suite of development tools and services that facilitates end-to-end application development and delivery.

During mentoring sessions, we learned valuable skills on using Azure DevOps for efficient collaboration and version control. The mentors guided us through essential processes, such as cloning projects, creating branches, and initiating pull requests, enabling seamless teamwork and code integration. Additionally, Azure DevOps provides robust work item tracking, allowing us to manage and prioritize tasks effectively.

The mentors demonstrated how to leverage these work items to keep track of project progress and coordinate efforts within a team. The integration with Visual Studio was particularly valuable, showcasing how to fetch and pull changes effortlessly, ensuring that our local environments were always synchronized with the latest updates made to the project.

This hands-on experience with Azure DevOps significantly streamlined our development workflow, emphasizing best practices in version control and collaboration within a modern software development environment.

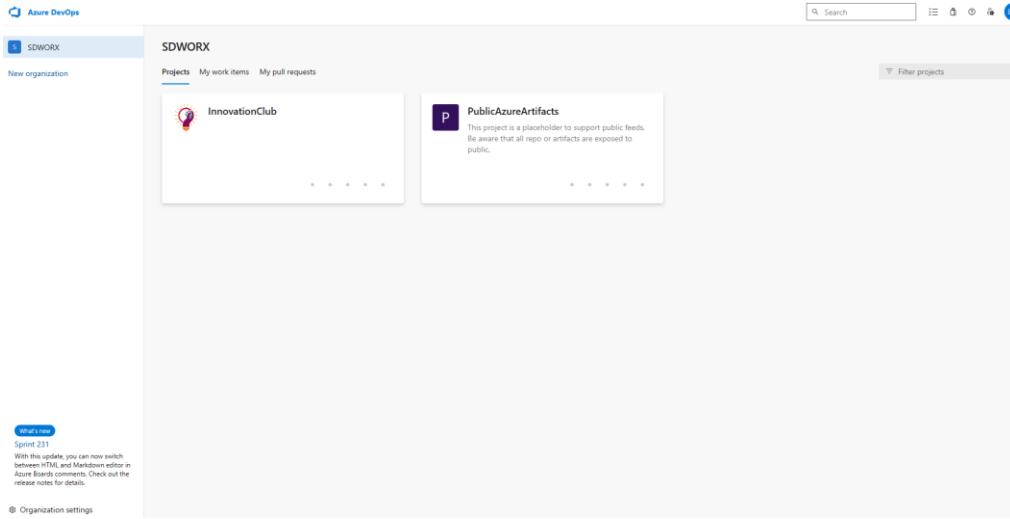


Figure 10 shows Azure DevOps Homepage

2. Quality Assurance Testing

Our mentors in the QA testing domain provided invaluable training, imparting essential skills in Quality Assurance (QA) testing to our team. They guided us through the intricacies of both manual testing and automation testing methodologies. In the realm of manual testing, our mentors shared insights into crafting effective test cases, executing comprehensive test scenarios, and documenting results.

They emphasized the importance of thorough test coverage to ensure the reliability and functionality of software applications. Additionally, our mentors introduced us to automation testing tools and frameworks, elucidating the benefits of automating repetitive test cases to enhance efficiency and speed up the testing process.

It involved meticulously testing various user interactions to identify potential issues and bugs. Through a systematic approach, I executed a range of test scenarios, scrutinizing the website's performance under different conditions. This encompassed assessing the platform's responsiveness, evaluating user interface elements, and confirming the accuracy of data processing.

Hands-on sessions allowed us to gain proficiency in writing and executing automated test scripts, integrating testing seamlessly into our continuous integration workflows. The mentorship not only equipped us with technical expertise but also instilled a quality-first mindset, emphasizing the critical role of QA testing in delivering robust and reliable software solutions.



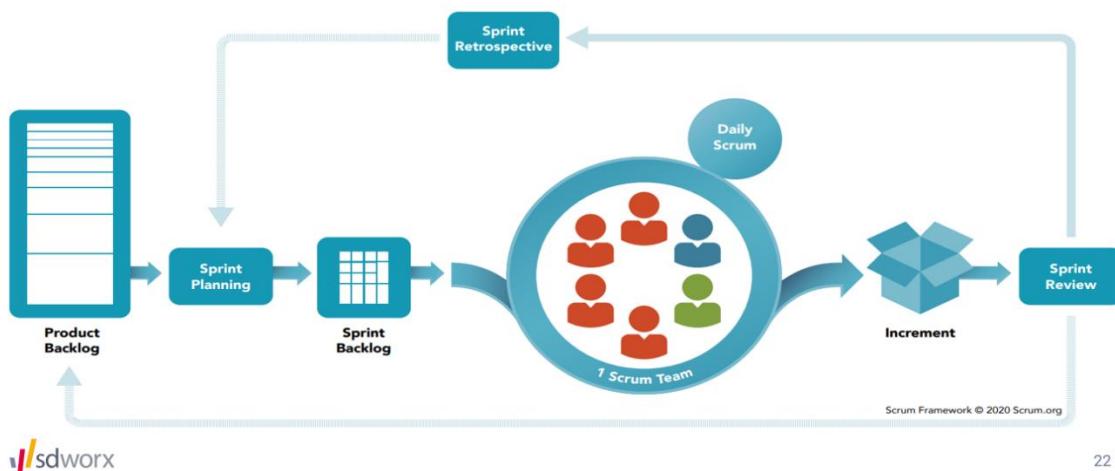
sdwrx

3

Figure 11 shows the Training Plan

Agile/ Scrum

SCRUM FRAMEWORK



sdwrx

22

Figure 12 shows the Agile/Scrum framework

Agile/Scrum is a widely adopted framework for project management, emphasizing iterative and collaborative development practices. Scrum provides a structured approach to managing complex projects by breaking them into smaller, manageable increments. The framework is guided by key principles outlined in Scrum Theory, emphasizing transparency, inspection, and adaptation. Scrum Values, including commitment, courage, focus, openness, and respect, form the foundation of a collaborative and empowered team environment. The Scrum Team consists of three roles: Developers, responsible for delivering the product increment; the Product Owner, accountable for maximizing product value; and the Scrum Master, who facilitates the Scrum process and ensures adherence to its principles.

Scrum Events, such as Sprint Planning, Sprint Review, and Daily Scrum, provide regular opportunities for collaboration and feedback within a time-boxed framework. The Sprint, a time-boxed iteration, enables the team to deliver a potentially shippable product increment. The Product Backlog serves as the prioritized list of features, enhancements, and bug fixes, while the Sprint Backlog details the tasks to be accomplished during the sprint.

The Scrum framework's iterative and adaptable nature makes it a good practice for managing projects with evolving requirements and uncertainties. It fosters teamwork, transparency, and continuous improvement, allowing organizations to respond swiftly to changing market demands and deliver high-quality products. Overall, Scrum provides a structured yet flexible approach that aligns the development team, stakeholders, and product goals, promoting a collaborative and efficient development process.

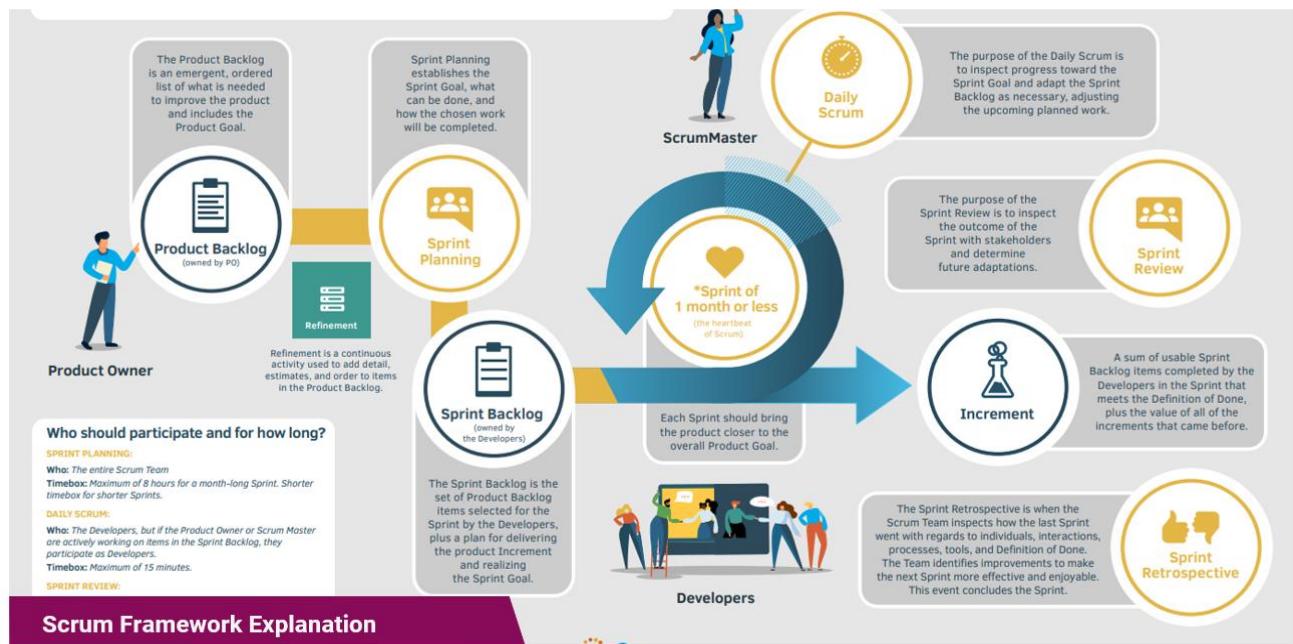


Figure 12 shows the Scrum Framework Breakdown

Manual Testing

During the training sessions, our mentors assigned exercises that specifically focused on the equivalence portioning, boundary value analysis and creating test cases for diverse scenarios where users interacted with a website. These exercises served as practical applications of our learning in quality assurance testing. Guided by the mentors, I engaged in the systematic formulation of test cases, considering different user journeys and potential interactions within the application.

The exercises encouraged a hands-on approach to understanding the intricacies of creating comprehensive test scenarios, covering aspects such as positive and negative test cases, boundary testing, and validation of user inputs. By actively participating in these exercises, I not only honed my skills in crafting effective test cases but also gained a deeper appreciation for the critical role of thorough testing in ensuring the reliability and functionality of software applications.

The mentor-led exercises were instrumental in translating theoretical knowledge into practical expertise, fostering a proactive and meticulous approach to quality assurance testing.

Exercise 1

Equivalence and Boundary Value

- Let's consider the behaviour of Order Pizza Text Box Below
- Pizza values 1 to 10 is considered valid. A success message is shown.
- While value 11 to 99 are considered invalid for order and an error message will appear, "**Only 10 Pizza can be ordered**".

Here is the test condition.

1. Any Number greater than 10 entered in the Order Pizza field (let say 11) is considered invalid.
2. Any Number less than 1 that is 0 or below, then it is considered invalid.
3. Numbers 1 to 10 are considered valid.
4. Any 3 Digit Number say -100 is invalid.

Draw the test conditions with specify the test conditions separately for Equivalence and boundary value.

Exercise 2

To join the ABCD Members Club, you have to be an employee of SD Worx between 18 and 65; anyone under 30 gets a 15% discount; anyone over 50 gets a 10 % discount.

- a) Which test techniques are you going to use?
- b) Identify all valid and all invalid ranges

Figure 13.1, Questions provided

Esae Moos

Exercise 1

Equivalence Partitioning

Invalid	Valid	Invalid
-100	0	1

Boundary Value

Invalid Test Case	Valid Test Case	Invalid Test Case
0	1,2,9,10	11

Exercise 2

(a), Boundary Value

b),

Valid:

18 <= age <= 65 : to be an employee
18 <= discount <30 : get a 15% discount
50 < discount <=65 : get a 10% discount

Invalid:

x < 18 OR x > 65 : not be an employee of SD Worx
x < 18 OR x >= 30 AND x< 50 : not get the 10% discount
x < 50 : to not get the 15% discount
30 <= x <= 50 : no discount

Figure 13.2, Corrections

User Story:

As the administrator of the OrangeHRM Website, I need the capability to both add and view Employment Status in the admin section.

The required information to establish Employment Status includes the following:

Employment status field

Environment Details:

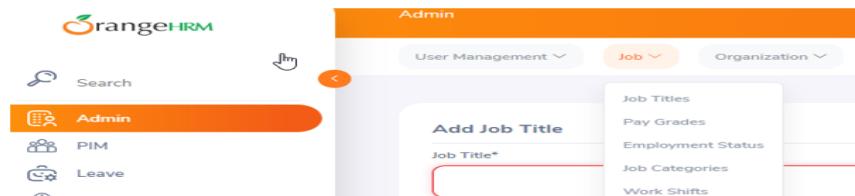
OrangeHRM Website: orangehrmlive.com

Credentials:

Username: Admin

Password: admin123

To access this feature, follow the provided credentials and navigate to Admin Section > Job.



Acceptance Criteria

- Given I am an administrator

When I add an employment status in the system

Then the employment status should be successfully added

Figure 14.1, User Story Case Scenario

2. Employment status with more than 50 characters (positive scenario)

Acceptance Criteria

- Given I am an administrator,
When I create an employment status with more than 50 characters,
Then the system should not allow the employment status to be created,
And I should see a warning message indicating the character limit,

Test Scenario ID	Add_EmpStatus50	Test Case ID	Add_EmpStatus50
Test Case Description	Adding a new Employee Status with name exceeding 50 characters - positive case scenario	Test Priority	High
Pre-Requisite	A valid admin user account	Post-Requisite	NA

Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Shared steps		Shared steps completed successfully	Shared steps completed successfully	Pass	Launch Successful
6	Go to Add Employment Status page	Click Add button	Add Employment Status page	Add Employment Status page	Pass	Add Employment Status page opens successfully
7	Input Name	Valid Name: 'khfjdhfkjdsh fkjshfdkjhieii wiwwwfergfg fdvcvxcvds gfs'	Error message displayed 'Should not exceed 50 characters'	Error message displayed 'Should not exceed 50 characters'	Pass	Name validated

Figure 14.2, User Story Case Scenario answers

Playwright (Automation Testing)

I was also given the training for Playwright, a Node.js library that lets you script and automates browsers using the same API, like Chrome, Firefox, and Safari. This cross-browser automation is evergreen, capable, reliable, and fast! I was given detailed explanation and tutorials on such concept in a YouTube Playlist, made by a coding youtuber



Figure 15.1, Automation Playwright YouTube video

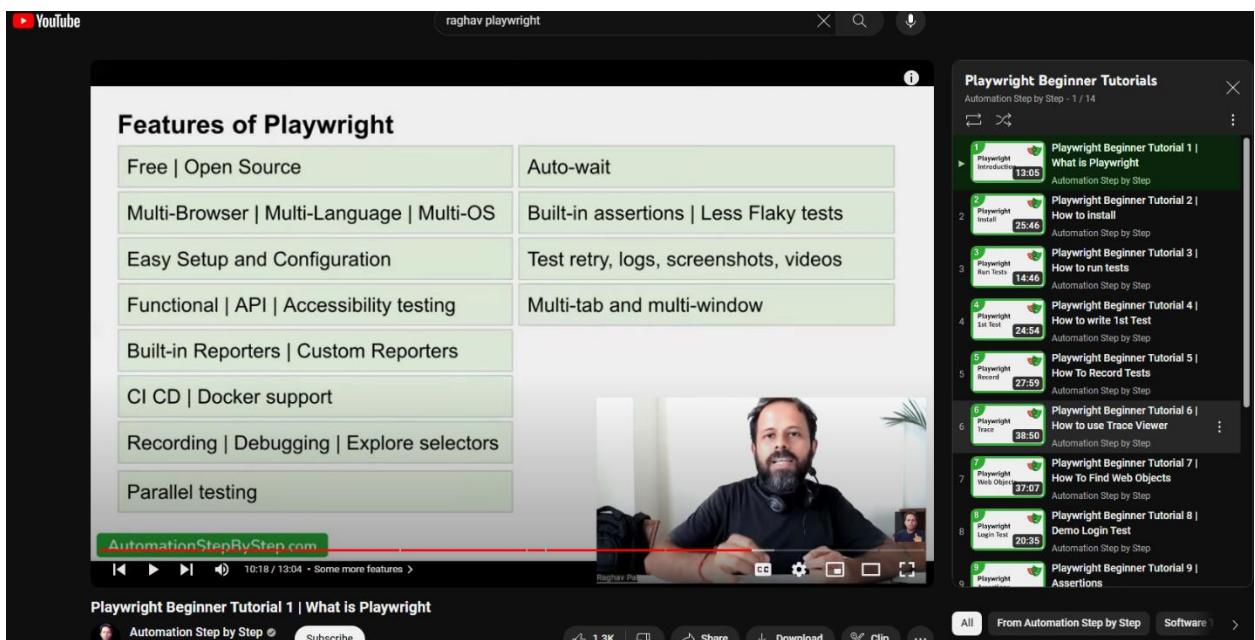


Figure 15.2, Automation Playwright YouTube video features

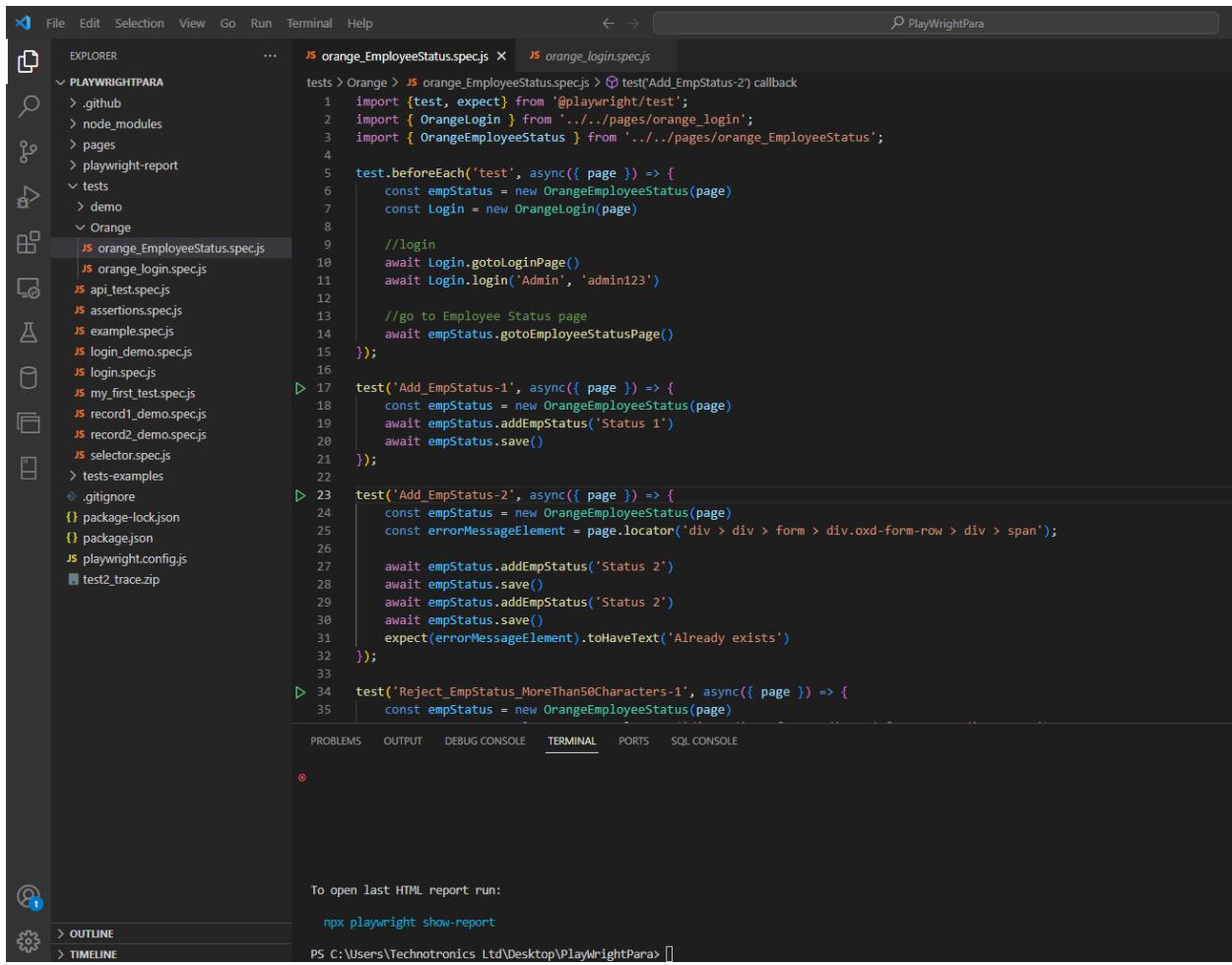
Factors of Playwright that I learnt is:

- How to install
- How to run and record tests
- How to use Trace Viewer
- How to find Web Objects
- Hooks, Groups and Assertions
- Annotation and Tags
- Object Model

With the help of these YouTube tutorials, finding valuable resources that provided step-by-step guidance on understanding Playwright's features and capabilities. Armed with this knowledge, I transitioned to practical application by automating various case scenarios using Playwright.

Visual Studio as my integrated development environment, I seamlessly integrated Playwright into my workflow, enabling me to write and execute automation scripts efficiently. This hands-on experience not only solidified my understanding of Playwright's syntax and functionalities but also empowered me to automate complex user interactions with ease.

The combination of self-directed learning through online tutorials and practical implementation in Visual Studio proved to be a dynamic and effective approach, allowing me to harness the full potential of Playwright for robust and efficient test automation.



The screenshot shows the Visual Studio interface with the following details:

- File Explorer:** Shows the project structure under "PLAYWRIGHTPARA". The "orange_EmployeeStatus.spec.js" file is selected, and its content is displayed in the main editor area.
- Main Editor:** Displays the Playwright test code for "orange_EmployeeStatus.spec.js". The code includes imports for playwright/test, OrangeLogin, and OrangeEmployeeStatus, and defines several test cases for adding employee status and handling validation errors.
- Bottom Bar:** Shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SQL CONSOLE. The TERMINAL tab is active.
- Terminal:** Shows the command "npx playwright show-report" and the path "PS C:\Users\Techntronics Ltd\Desktop\PlayWrightPara\".
- Bottom Left:** Shows icons for OUTLINE and TIMELINE.

Figure 16, Playwright codes in Visual Studio

4.0 Implementation

Description of Internship Assignment - SDEvent

During my internship, one of my primary assignments was the development of a comprehensive calendar system named SDEvents. This project aimed to streamline event management within the organization by providing administrators with a user-friendly interface to create, update, and manage upcoming events. SDEvents empowered administrators with the ability to effortlessly inform users about scheduled events, fostering effective communication within the team. The platform featured robust functionality, allowing administrators to edit user profiles and event details with ease. A notable highlight of SDEvents was its calendar option, providing users with a visual representation of scheduled events, complete with date and time details. This feature not only enhanced the user experience but also contributed to efficient planning and coordination within the organization.

Azure DevOps, our main project management environment, played a central role in our collaborative development process, serving as a comprehensive platform for project management and version control. In the initial stages, our mentors utilized Azure DevOps to create tasks and assignments for the team, fostering a structured and organized approach to project development. As the project progressed, I assumed the role of Scrum Master, and the responsibilities evolved. I became the point of contact for task assignments, using Azure DevOps to allocate specific responsibilities to team members. Within Azure DevOps, all tasks for each team member were meticulously listed, providing transparency and accountability for the entire team.

Branching became an integral aspect of our development workflow. Each team member created their own branches to work on individual tasks, ensuring a parallel development process without affecting the main project. After completing the assigned tasks, the next step involved creating pull requests. Azure DevOps facilitated this process, allowing us to submit our changes for review by the mentors. Pull requests were meticulously managed, providing an opportunity for code reviews, discussions, and a controlled integration of changes into the master branch.

This structured approach ensured a seamless and collaborative development environment, enhancing our efficiency and the overall quality of the project. Azure DevOps proved to be an indispensable tool, orchestrating our tasks, branches, and pull requests seamlessly within the Agile/Scrum framework.

The screenshot shows the Azure DevOps Work Items page for the InnovationClub board. The main area displays a list of work items with the following columns:

ID	Title	Assigned To	State	Area Path	Tags	Comments	Activity Date
814108	As an admin, I wish to be able to deactivate the account of a ...	Shivani Ramalingum	Active	InnovationClub	Event Planner		1/5/2024 1:08:47 PM
814919	As a user, I wish to see a better home page	Shivani Ramalingum	Active	InnovationClub	Event Planner		1/5/2024 1:08:11 PM
808163	As a user, I want details of an event clicked on the calendar to be d...	Esaie Moos	Active	InnovationClub	Event Planner		12/27/2023 12:37:26
808166	As a admin, I wish to be able to update and/or deactivate/freeze a ...	Shivani Ramalingum	Active	InnovationClub			12/20/2023 6:17:51 AM
808139	As a backend engineer, I want to update EventType by EventCateg...	Jeebodhun Abhisarika	Active	InnovationClub	Event Planner		12/20/2023 6:12:40 AM
800578	As a user, I wish that my data is loaded when I logon	Yusra Cader	Active	InnovationClub	Event Planner		12/20/2023 6:08:57 AM
808143	As a user, I wish that my search result are updated in real-time	Kalim Auckbaraulie	Active	InnovationClub	Event Planner		12/19/2023 7:23:04 AM
808126	As a UI/UX Engineer, I need to fix various aspects of the visuals of t...	Manav Rammrarin	Active	InnovationClub	Event Planner		12/19/2023 6:24:06 AM
800560	As an admin, I should be able to see user data	Unassigned	New	InnovationClub	Event Planner		12/18/2023 8:39:43 AM
801149	As an engineer, I should be able to link events to users.	Manav Rammrarin	Resolved	InnovationClub	Event Planner		12/11/2023 7:54:14 AM
800464	As a user, I wish navigating through the app to be user friendly	Yusra Cader	Resolved	InnovationClub		4	12/11/2023 6:01:22 AM
667782	Calendar of Events	Khilesh Chumbit	Active	InnovationClub	Event Planner	2	11/27/2023 8:13:25 AM
700903	Manage Events	Unassigned	Resolved	InnovationClub	Event Planner	2	11/24/2023 7:28:13 AM
700895	Admin Home Page	Unassigned	Resolved	InnovationClub	Event Planner		11/24/2023 7:27:59 AM
799450	As a DEV, I want my code to be well documented	Ridhwaan Paraouty	Resolved	InnovationClub			11/23/2023 1:31:04 PM
783560	As a user, I should be able to subscribe to an event.	Unassigned	New	InnovationClub	Event Planner		10/16/2023 2:39:48 PM

Figure 17, Azure DevOps Work Items

All the work items for all team members are listed in the Work Items page. The work item is given and ID as a unique identifier so as we can find it again alter on. It has a title to give a description on what the work is about. It also shows who it is assigned to and the state of the

The screenshot shows the detailed view of a work item (Task 808164) in the Azure DevOps Work Items page. The work item details are as follows:

- Title:** TASK 808164
- Description:** 808164 - DEV: Display event details in a drawer when a user clicks on an event.
- Assignee:** Esaie Moos
- Comments:** 0 comments
- Planned:** Priority: 2, Activity: Development
- Effort (Hours):** Original Estimate: Remaining: Completed: 0
- Implementation:** Integrated in Build
- Deployment:** To track releases associated with this work item, go to Releases; and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.
- Development:** + Add link
- Related Work:** + Add link
- External Link:** Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
- Source Reference:** Source Reference

work such as Active, if the work is still ongoing or Resolved if the user has already worked and finished implementing it.

Figure 18, Task item description

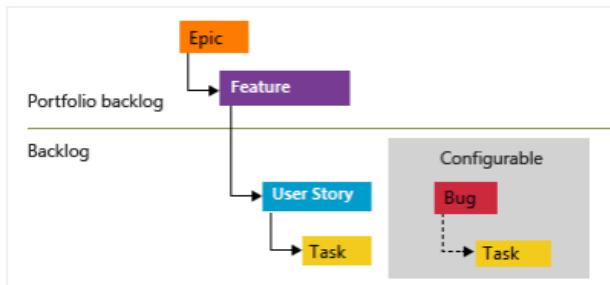
The screenshot shows the Azure DevOps interface for a work item titled "USER STORY 808163". The left sidebar is the Boards hub, with "Work items" selected. The main area displays the following details:

- Description:** As a user, I want details of an event clicked on the calendar to be displayed in a drawer.
- Acceptance Criteria:** Click to add Acceptance Criteria.
- Discussion:** Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.
- Planning:** Story Points: 2, Priority: 2, Risk: N/A.
- Deployment:** To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.
- Development:** Display Drawer (Created 12/27/2023, Active).
- Related Work:** Add link, Add an existing work item as a parent.
- External Link:** ReflectedWorkItemID.
- Environment:** Found In Environment.

Figure 19, Task User Story

Azure Boards

Agile Process



The above image depicts an agile process of a backlog of a team. SDEvents uses the same process. Epic and Features are the main containers that defines the backbone of a project using the agile process. An epic can house many features, features can house many user stories, and user stories can house **many** tasks (and many people can be assigned these tasks, e.g. Abhisarika on a DEV task, Manav on a QA task), and so on.

Example: SDEvents

- **Epic:** SDEvents
- **Feature:** SDEvents -> Homepage
- **User Story:** SDEvents -> Homepage -> 301952 As a User, I wish to have a homepage
- **Task:** SDEvents -> Homepage -> US301952 -> 301953 DEV: Implement a homepage
- **Task:** SDEvents -> Homepage -> US301952 -> 301956 QA: Implement a homepage
- **Bug:** SDEvents -> Homepage -> 301954 Client's name is not populated on homepage
- **Task(BUG):** SDEvents -> Homepage -> BUG301954 -> 301955 DEV: Fix homepage not populating name

Figure 20, Azure Boards Agile Process

Anatomy of an Item

Using a User Story

- Item number – 301952(example)
- Item Title – ‘As a user, I wish to have a homepage’
- Assigned to/ esaiemoos420@gmail.com – The team member to which the item (User Story in this case) is assigned to. In this case, I would be working on this user Story.
- State – The states of the work item.
 - New: No one is working on that particular work item currently.
 - Active: It is assigned to someone and is currently being worked on.
 - Resolved: The previous person who was working on it has completed their task assigned. Someone else who has a task in this user story can work on it.
 - Closed: This work item doesn’t need any more work. It is closed unless a change needs to be done.
- Tags – Can be used to categorise work items.

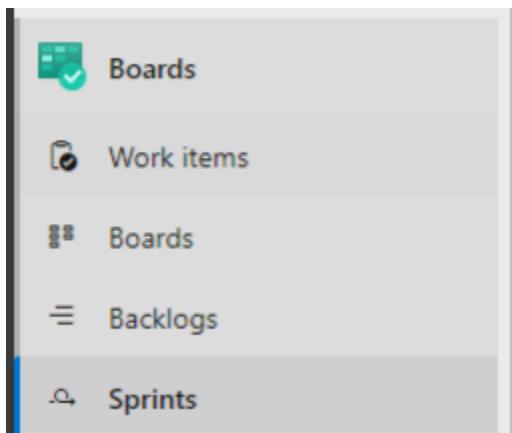
Using a Task

Description: Description of the task, what needs to be done.

Discussion: Updates about this task. When the user assigned has completed a Pull Request, generated comments are found and their changes were successfully merged with the master branch.

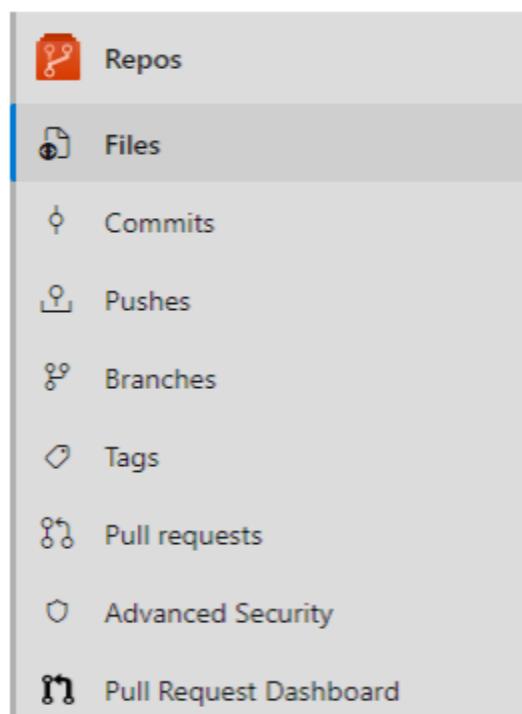
Planning -> Activity: Needs to be updated to the appropriate activity. Navigate to one of the tasks and see the different available activities.

Effort (Hours): This is typically set by the Scrum Master and is an estimate of how much this task will take.



In the **Boards Section**, These 4 can be used to see tasks that are assigned to me, and even assigned to my teammates.

It also houses unassigned tasks and others.



The **Repos Tab**, where files reside.

Commits: History of commits pushed to the repo. For example, I worked on a login feature and committed a change named 'Implemented Login Feature'

Pushes : History of pushes sent to the repo.

Branches: Where branches can be managed or viewed.

Pull Request: List of pull requests from a branch to another branch (master).

For example, the request to pull changes from branch "feature/303457-implement-login-page" into the "master" branch. So that commit is merged with the "master" branch therefore making master up to date.

SDEvents

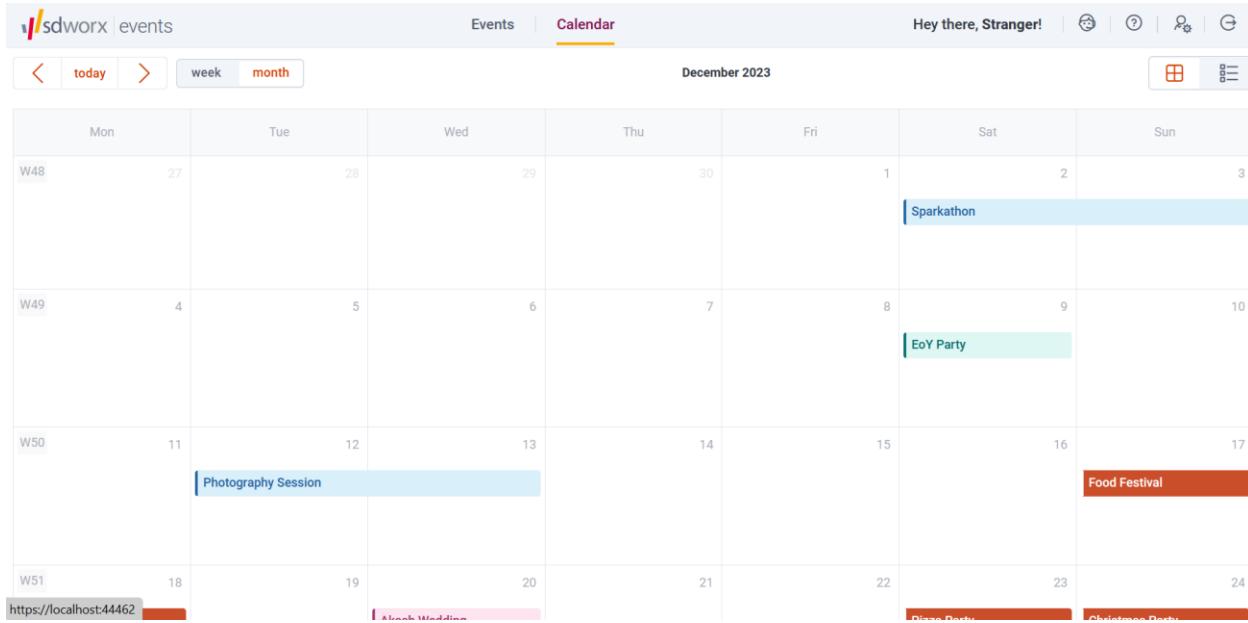


Figure 21, SDEvents, Calendar Page

For the homepage, categories “All Events” and “Upcoming Events” were listed on top of another. I was tasked with displaying “All Events” and “Upcoming Events” side-by-side where I used a concept called Bootstrap Grid System, which I was able to complete in the first 2 days upon being assigned to it. I created my own branch called event-list and committed and push the changes to the master branch

```

1 <div class="container-fluid">
2   <h1 class="mb-0">Events</h1>
3   <hr class="mt-0" />
4   <button ngbTooltip="Events" placement="bottom" [routerLink]="/event-form" class="btn btn-secondary mb-2">
5     | <i class="ionicons8-o-plus"></i>
6     | <span>Add Event</span>
7   </button>
8
9   <div class="d-flex justify-content-between my-3">
10    <div class="flex-grow-1 me-2">
11      <input class="form-control" type="search" placeholder="Search" aria-label="Search" name="search" [(ngModel)]="searchModel" />
12    </div>
13    <div class="flex-shrink-1 me-2">
14      <select class="form-select mb-2" id="eventTypeFilter" [(ngModel)]="selectedEventType" (click)="filterDropdownClicked = true">
15        <option *ngIf="!selectedEventType" value="">Filter by Type</option>
16        <option value="brunch">Brunch</option>
17        <option value="hackathon">Hackathon</option>
18      </select>
19    </div>
20  </div>
21
22  <div class="row">
23    <div class="col-md-6">
24      <div class="card shadow">
25        <div class="card-header">
26          <h3 class="mb-0">All Events</h3>
27        </div>
28        <div class="card-body">
29          <table class="table table-hover table-borderless">
30            <thead class="thead-light">
31              <tr>
32                <th>Event Name</th>
33                <th>Date</th>
34                <th>Location</th>
35                <th>Status</th>
36                <th class="text-center">Actions</th>
37              </tr>
38            </thead>
39            <tbody>
40              <tr *ngFor="let event of filteredEvents" class="align-middle">
41                <td>{{ event.name }}</td>
42                <td>{{ event.startDateTime | date: 'E, MMM d, y \'at\' h:mm a zzzz' }}</td>
43                <td>{{ event.location.name }}</td>
44                <td>{{ setEventStatus(event.status) }}</td>
45                <td class="text-center">
46                  <div class="d-flex justify-content-center">
47                    <button class="btn btn-link">
48                      | <i class="ionicons8-o-trash"></i>
49                    </button>
50                    <button type="button" (click)="updateRSVP(event.id)" class="btn btn-link">
51                      | <i [ngClass]="'isGoing ? 'ionicons8-o-ok' : 'ionicons8-o-cancel'"></i>
52                    </button>
53                  </div>
54                </td>
55              </tr>
56            </tbody>
57          </table>
58        </div>
59      </div>
60    </div>
61  </div>
62

```

Figure 22.1, event-list html codes

```

58  <div class="card shadow">
59    <div class="card-header">
60      <h3 class="mb-0">Upcoming Events</h3>
61    </div>
62    <div class="card-body">
63      <table class="table table-hover table-borderless">
64        <thead class="thead-light">
65          <tr>
66            <th>Event Name</th>
67            <th>Date</th>
68            <th>Location</th>
69            <th>Status</th>
70            <th class="text-center">Actions</th>
71          </tr>
72        </thead>
73        <tbody>
74          <tr *ngFor="let event of filteredEvents" class="align-middle">
75            <td>{{ event.name }}</td>
76            <td>{{ event.startDateTime | date: 'E, MMM d, y \'at\' h:mm a zzzz' }}</td>
77            <td>{{ event.location.name }}</td>
78            <td>{{ setEventStatus(event.status) }}</td>
79            <td class="text-center">
80              <div class="d-flex justify-content-center">
81                <button class="btn btn-link">
82                  | <i class="ionicons8-o-trash"></i>
83                </button>
84                <button type="button" (click)="updateRSVP(event.id)" class="btn btn-link">
85                  | <i [ngClass]="'isGoing ? 'ionicons8-o-ok' : 'ionicons8-o-cancel'"></i>
86                </button>
87              </div>
88            </td>
89          </tr>
90        </tbody>
91      </table>
92    </div>
93  </div>
94

```

Figure 22.2, event-list html codes

Then I was tasked with implementing a drawer function in the calendar page. When an event is clicked by a user, a drawer opens up showing some information about that particular event. This was done simply by implementing codes, TypeScript and HTML.

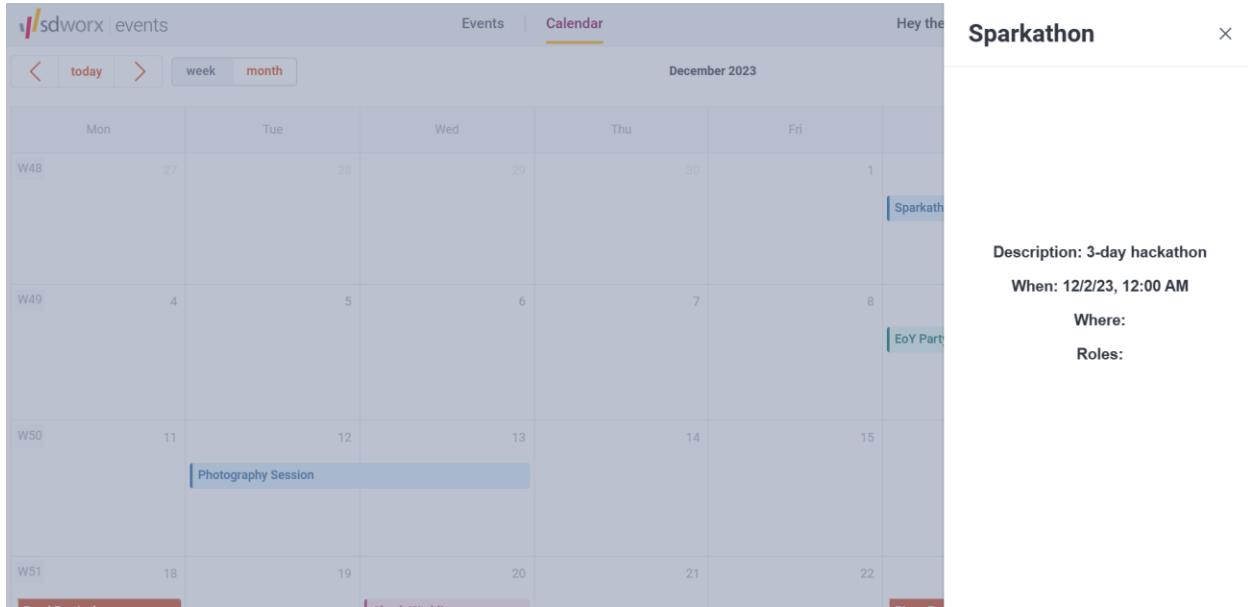


Figure 23.1, Drawer function in calendar

```

src > app > shared > components > drawer > TS drawer.component.ts > DrawerComponent
You, 2 weeks ago | 1 author (You)
1 import { Component, Input, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-drawer',
5   templateUrl: './drawer.component.html',
6   styleUrls: ['./drawer.component.scss'],
7 }
8 export class DrawerComponent implements OnInit {
9   @Input() eventData: any;
10
11   ngOnInit(): void {
12     | console.log(this.eventData);
13   }
14 }
15

```

Figure 23.2, Drawer function TypeScript code

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under 'CLIENTAPP' with files like calendar.component.css, event-form.component.html, events-list.service.ts, etc.
- Editor:** Displays the HTML code for 'drawer.component.html'. The code includes styles for a drawer body, an offcanvas header with a title and close button, and a body containing event details such as description, when, where, and roles.
- Terminal:** Shows the message "Compiled successfully."
- Output:** Shows a warning from 'src/styles.scss' about mixed support for 'end' values.
- Bottom Status Bar:** Includes icons for node.js, terminal, file, and more.

```

1 <style>
2   .drawer-body{text-align: center;}
3   .text-div {
4     line-height:50px;
5     margin-top: 125px;
6     font: bold 15px "Helvetica Neue", Helvetica, Arial, sans-serif;
7   }
8 </style>
9
10 <div class="drawer">
11   <div class="offcanvas">
12     <div class="offcanvas-header">
13       <h4 class="offcanvas-title">{{eventData.title}}</h4>
14       <button type="button" class="btn-close" data-bs-dismiss="offcanvas" aria-label="Close"></button>
15     </div>
16     <div class="offcanvas-body">
17       <div class="drawer-body">
18         <div class="text-div">
19           <p>
20             | <span class="fw-bold">Description: </span>{{ eventData.extendedProps.description }}
21           | <span class="fw-bold">When: </span>{{ eventData.extendedProps.startTime | date : "short" }}
22           | <span class="fw-bold">Where: </span>{{ eventData.extendedProps.locations[0].Address }}
23           | <span class="fw-bold">Roles: </span>{{ eventData.extendedProps.roles[0].EventRole }}
24         </div>
25       </div>
26     </div>
27   </div>
28 </div>
29
30
31
32

```

Figure 23.3, Drawer function HTML code

After successfully implementing and testing my code changes within Visual Studio Code, the next step in the development process involved committing and pushing these changes to the master branch. Leveraging the integrated version control features of Visual Studio Code, I staged my modifications, providing a clear snapshot of the changes made.

The commit process allowed me to encapsulate a logical set of changes along with descriptive commit messages, facilitating easy understanding for collaborators and future reference. With that completed, I pushed the changes to the master branch directly from Visual Studio Code, ensuring that my contributions were integrated into the main project repository. This streamlined workflow within my development environment not only expedited the process but also maintained the integrity of the project's version control, reflecting a disciplined and collaborative approach to software development.

The screenshot shows the Visual Studio Code interface. On the left, the Source Control sidebar displays a commit history for a project named 'ClientApp'. One commit is selected, showing the code for 'DrawerComponent'. The code implements the `OnInit` interface and logs event data. Below the commit history, the file tree shows various Angular components and services. The bottom right pane shows the terminal output, which includes a warning from PostCSS loader about mixed support for the `end` value in a CSS autoprefixer rule.

```

    import { Component, Input, OnInit } from '@angular/core';
    ...
    export class DrawerComponent implements OnInit {
        @Input() eventData: any;
        ...
        ngOnInit(): void {
            console.log(this.eventData);
        }
    }

```

Figure 24.1, Commit function in Visual Studio

The screenshot shows the 'Commits' page in the Azure DevOps interface for the 'InnovationClub' repository. The left sidebar has 'Commits' selected. The main area displays a list of merged pull requests (PRs) in a graph format. Each PR entry includes the author's name, date, and a link to the PR details. The commits are:

- Merged PR 94244: Load current user data using User service
- Merged PR 93844: Implemented roastr and changed date ui
- Merged PR 93750: fixed visual aspects
- Merged PR 93728: update-search-bar
- Merged PR 93578: /changes to date time and added cards in events
- Merged PR 93549: Merged the codes for Update and Create events into a single component
- Merged PR 93183: Branch Fix
- Merged PR 93178: Fixed Incorrect Import
- Merged PR 91796: Put some comments + created a user service
- Merged PR 91795: Modify the navbar to separate the main navigation
- Merged PR 91799: Change layout of Event
- Merged PR 91800: Implemented Event rsvp UI
- Merged PR 91852: Implemented userevent Service

Figure 24.2, Commits page made by teammates

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with a tree view of the database structure, including tables like 'sdevents', 'EventCategories', and 'Roles'. The right pane shows a table named 'Events' with the following data:

EventId	Location...	Name	StartDateTime	EndDateTime	Description	Status	Scope	Category...
398784a090	8650e11...	Photography Session	2023-11-12 00:00:00	2023-11-14 00:00:00	Employee photo session	1	2	Hackathon
070fd86...	b8a568b...	Tii Alexandre Concert	2023-11-27 00:00:00	2023-11-29 00:00:00	Tii Alexandre Music Festival2	0	3	Hackathon
00a64e5...	8b17b8c...	EoY Party	2023-11-09 00:00:00	2023-11-10 00:00:00	End of Year Party	2	1	Hackathon
6bc46ec...	d9bde39...	Khilesh Party	2023-11-25 00:00:00	2023-11-26 00:00:00	Khilesh Party	2	1	Hackathon
3ad9aa7...	e8a74c7a...	Christmas Party	2023-11-24 00:00:00	2023-11-24 00:00:00	Christmas celebration in the office	1	2	Hackathon
92b7bb2...	b817b8c...	SD Pool Tournament	2023-11-25 00:00:00	2023-11-30 00:00:00	The SD Pool Tournament is an ann...	0	0	Hackathon
53dbdd9...	9e34d67...	Pizza Party	2023-11-23 00:00:00	2023-11-24 00:00:00	Ridwaan paying pizza	1	2	Hackathon
f018529...	8650e11...	Yogesh Party	2023-11-25 00:00:00	2023-11-26 00:00:00	Yogesh Party	2	1	Hackathon
c18b538...	d9bde39...	Sparkathon	2023-11-02 00:00:00	2023-11-04 00:00:00	3-day hackathon	1	1	Hackathon
5e37fbe0...	b8a568b...	Food Festival	2023-11-17 00:00:00	2023-11-19 00:00:00	Sale of various food items	2	1	Hackathon
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 25.1, SQL server, Database for SDEvents storing Events data

A robust and organized storage approach was adopted using an SQL Server database. The data model incorporated essential fields such as event ID, location, date, status, and description to capture key attributes of each event. The SQL Server database provided a relational structure, allowing for efficient storage and retrieval of information. The event ID served as a unique identifier, ensuring each record was distinctly identifiable. The location, date, and description fields accommodated the specific details of each event, while the status field facilitated tracking the current state of an event (e.g., upcoming, ongoing, or completed).

SQL Server not only ensured data integrity through relational constraints but also enabled the implementation of complex queries and reporting functionalities. The choice of SQL Server as the underlying database management system contributed to a well-organized and scalable solution for managing and querying events data within the project.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with a tree view of the database structure, including tables like 'sdevents', 'Users', and 'EventCategories'. The right pane shows a table named 'MININT-1OQAO7J.sdevents - dbo.Users' with the following data:

User Id	Role Id	Email	Name	Is Active
tedf35d7a2f	ceebabd99-6825-4...	kalim.auckbarullee@com...	Kalim Auckbarullee	False
2fd8bb6...	ceebabd99-6825-4...	ridhwaan.paraouty@com...	Ridhwaan Paraouty	False
1a16b94...	e244ca56-91df-4...	esaiemoops@company.co...	Esaie Moos	False
ac70f2c7...	ceebabd99-6825-4...	yusra.cader@company.co...	Yusra Cader	False
54a9bb6...	ceebabd99-6825-4...	dharun.teemul@company...	Dharun Teemul	False
f4f0c0d3...	e244ca56-91df-4...	shivani.ramalingum@com...	Shivani Ramalingum	False
c0a803f1...	ceebabd99-6825-4...	abhisarika.jeebodhun@co...	Abhisarika Jeebodhun	False
ba1f59f4...	ceebabd99-6825-4...	manav.ramnarain@comp...	Manav Ramnarain	False
32d04b5...	ceebabd99-6825-4...	doorgesh.neetye@compa...	Doorgesh Neetye	False
4a46bbd...	e244ca56-91df-4...	khilesh.chumbit@compan...	Khilesh Chumbit	False
*	NULL	NULL	NULL	NULL

Figure 25.2, SQL server, Database for SDEvents storing User Data

The storage of user data in the project also was structured within an SQL Server database. The data model encompassed essential fields such as user ID, role ID, email, and name to comprehensively capture user information. The user ID served as a unique identifier for each user, facilitating seamless record identification. The role ID allowed for the association of users with specific roles, defining their permissions and access levels within the system. User email and name fields stored personal information, ensuring a comprehensive representation of each user. SQL Server's relational database capabilities were leveraged to enforce data integrity through relationships, enabling efficient data retrieval and management.

The importance of SQL Server in this context lies in its ability to offer a secure, scalable, and relational storage solution. The relational database model facilitated the structured organization of data, ensuring the integrity and consistency of information related to events and users. SQL Server's support for complex queries and transactions allowed for efficient retrieval and manipulation of data, enhancing the overall performance of the system. Moreover, SQL Server's role in enforcing relationships and constraints contributed to maintaining data accuracy, preventing inconsistencies, and supporting referential integrity.

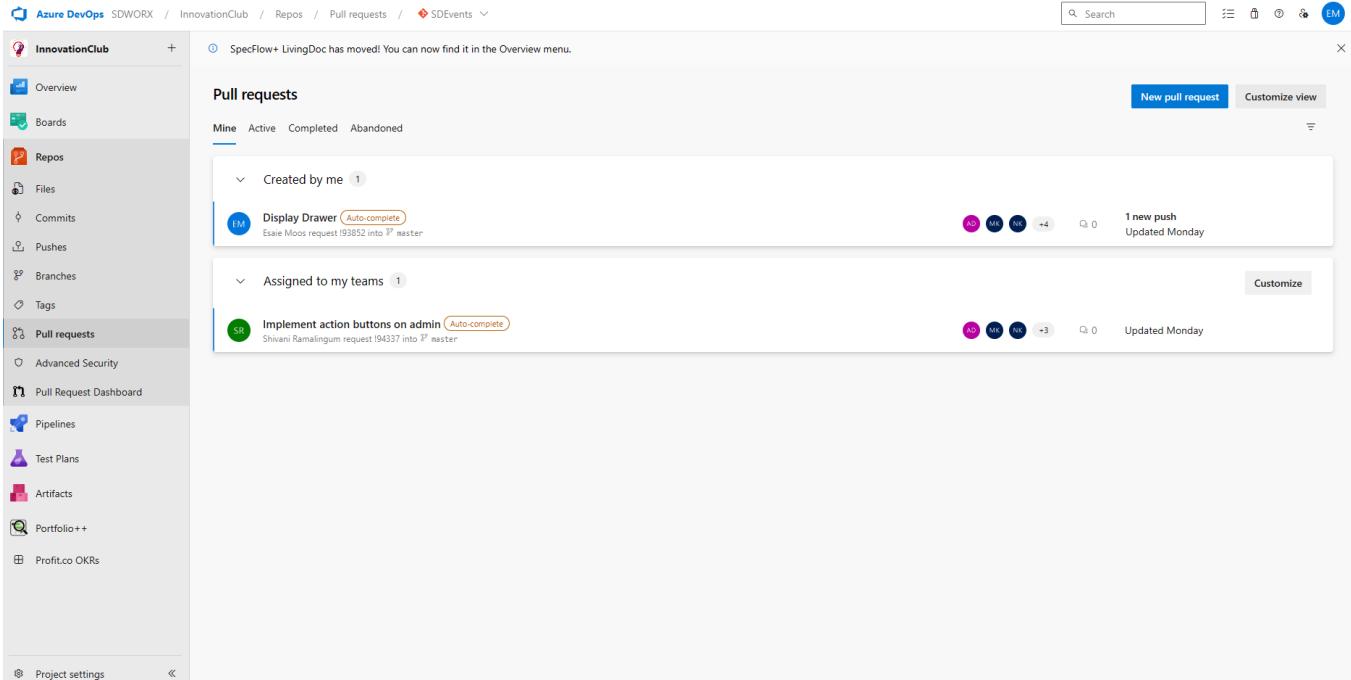


Figure 26.1, Pull Request Page 1

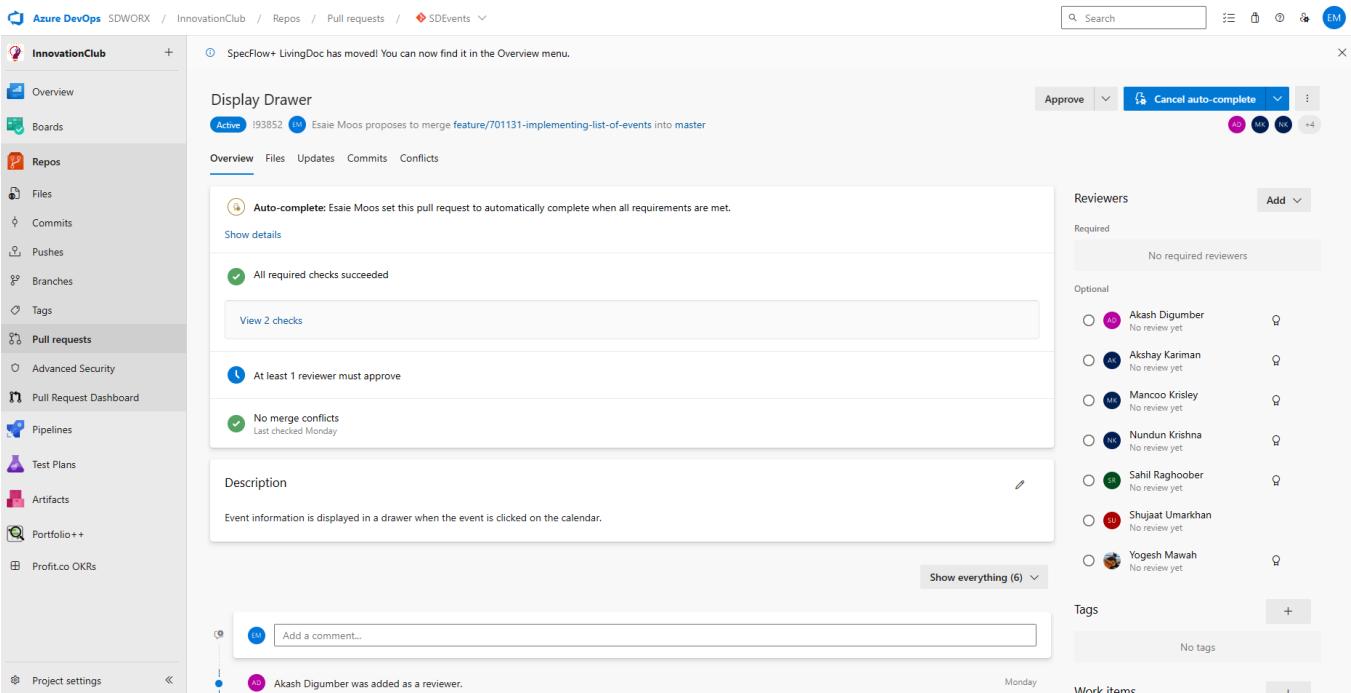


Figure 26.2, Pull Request Page 2

Right after I made the changes in SDEvents, I requested for a pull request so as the mentors can review my code and inspect for any error or check if it is working perfectly fine. SO then they can decide if they can approve the changes to the master branch, which will reflect new up to date changes to whoever is in the master branch.

The screenshot shows the Swagger UI for the SDEvents API version 1.0. At the top, there's a header with the Swagger logo and a dropdown menu labeled "Select a definition" set to "SDEvents.Api v1". Below the header, the title "SDEvents.Api 1.0 OAS3" is displayed, along with the URL "https://localhost:7070/swagger/v1/swagger.json".

EventCategories

- Events**
- EventCategories**

Events

- GET /api/Events/{eventId}** Gets an event with a specific ID
- POST /api/Events** Creates an event using eventRequest.
- GET /api/Events** Gets all events.
- PUT /api/Events** Updates an event using details from event request.

Figure 27.1, SDEvents API 1

The screenshot shows the Swagger UI for the SDEvents API version 2. It includes sections for "Location", "UserEvent", and "Users".

Location

- DELETE /api/Events/{EventId}** Deletes an event.
- POST /api/Events/status** Updates the status of an event.

UserEvent

- GET /api/UserEvent** Gets all events a user is interested in.

Users

- POST /api/Users**
- GET /api/Users**
- PUT /api/Users**

Figure 27.2, SDEvents API 2

The SDEvents API Swagger provides a clear and interactive interface for developers to explore and understand the capabilities of the API. Swagger serves as a powerful tool for describing the endpoints, operations, and parameters of the API in a standardized format, making it easier for developers to integrate and interact with the API. It not only lists all available endpoints but also provides detailed information on request and response formats, authentication methods, and any required parameters. It also allows developers to efficiently test API calls, understand expected outcomes, and seamlessly integrate the SDEvents API into their applications.

5.0 Other Events Organized by SDWorx

1. Hackathon

Dev Hackathon

In addition to our regular project work, we organized a **Mini Hackathon** that catered to both developers and QA testing. This unique initiative aimed to enhance our skills, foster creativity, and encourage collaboration within the team. The hackathon provided an opportunity for developers and QA testers to collectively contribute to innovative solutions and showcase their problem-solving abilities.

During the hackathon for developers, the challenge centered around building a personal task tracker application, emphasizing the integration of Dotnet 6 for the backend API and Angular for the frontend UI. I was tasked with developing a fully functional task management system, allowing for the seamless organization and tracking of individual tasks. This hands-on experience not only allowed developers to apply their knowledge of cutting-edge technologies but also promoted teamwork as participants collaborated on diverse aspects of the project, ranging from backend functionalities to the user interface. The hackathon not only served as a platform for skill enhancement but also fostered a sense of camaraderie and shared learning among the development team.

Features:

1. **Backend (Dotnet 6 API):**

- Create RESTful API endpoints for CRUD operations on personal tasks.
- Store tasks in-memory or use a simple database (e.g., SQLite) for persistence.
- Include endpoints for listing all tasks, adding a new task, updating an existing task, and deleting a task.

2. **Frontend (Angular):**

- Create a user interface with a list of tasks, displaying their titles and status.
- Include a form to add new tasks with a title and description.
- Implement functionality to mark tasks as complete or delete them.
- Add simple validation for the task form (e.g., required fields).

Workflow:

1. **Backend Development (Approx. 2-2.5 hours):**

- Set up a Dotnet 6 API project.
- Define a Task model with properties like id, title, description, and completed status.
- Implement API endpoints for CRUD operations on personal tasks.

- Test API endpoints using tools like Postman or Swagger.

2. Frontend Development (Approx. 2-2.5 hours):

- Set up an Angular project.
- Create components for displaying the list of tasks and the task form.
- Implement services to interact with the Dotnet 6 API endpoints.
- Develop the UI to list personal tasks, add new tasks, mark tasks as complete, and delete tasks.

3. Integration (Remaining Time):

- Integrate the Angular frontend with the Dotnet 6 API.
- Test the complete application to ensure seamless communication between the frontend and backend.
- Address any issues or bugs that arise during integration.

4. Presentation and Q&A (Final 15-30 minutes):

- Each intern presents their personal task tracker.
- Demonstrate the functionality, discuss design choices, and answer questions.

Personally, I was able to make 2 Task Tracker Applications which catered for the above requirements.

A screenshot of a modal window titled "Task Tracker". The window contains fields for "Task" (with placeholder "Add Task"), "Day & Time" (with placeholder "mm/dd/yyyy" and a calendar icon), and "Set Reminder" (with an unchecked checkbox). A large black button at the bottom right is labeled "Save Task". A red "Close" button is located in the top right corner of the modal.

Figure 28.1, Task Tracker 1

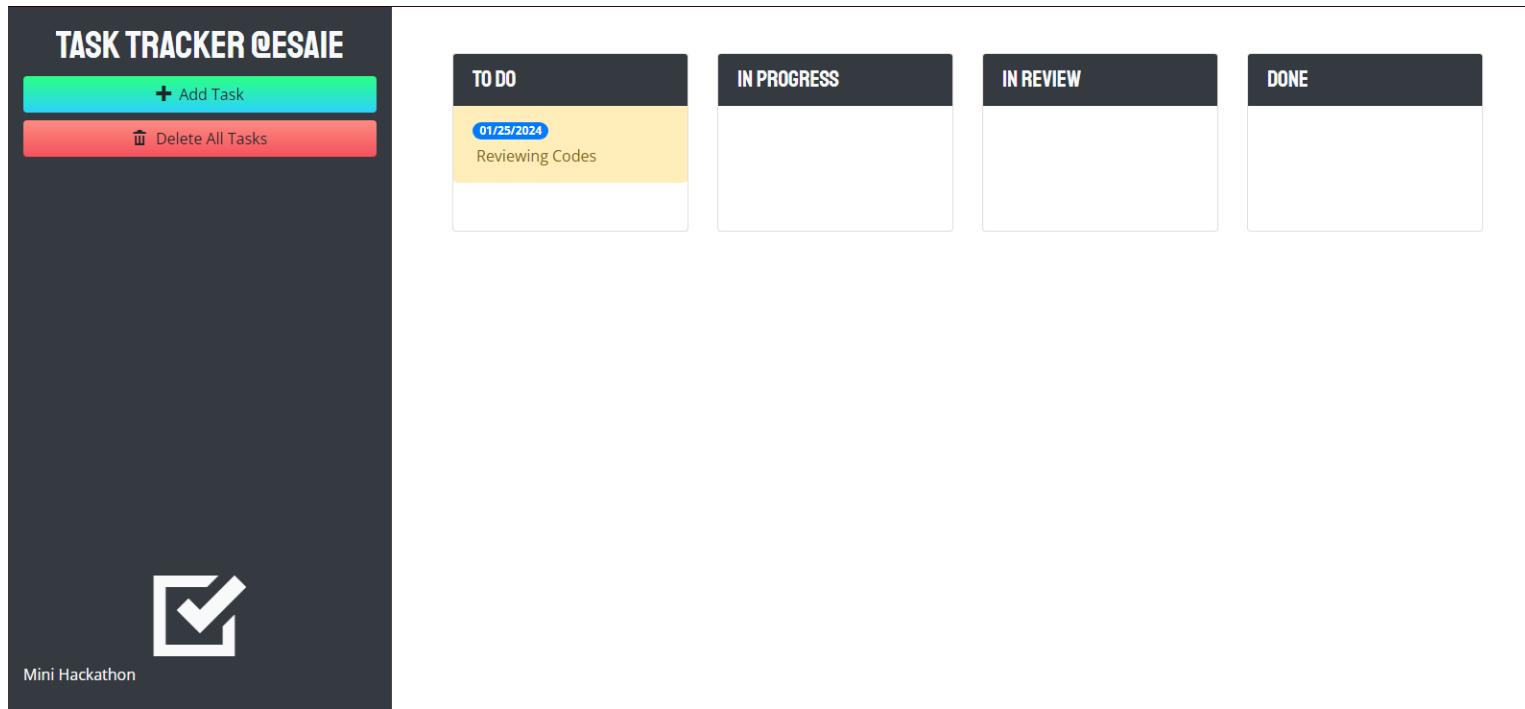


Figure 28.2, Task Tracker 2

QA Testing Hackathon

The QA testing hackathon provided a dynamic platform for assessing our skills and knowledge in various facets of quality assurance. The evaluators posed questions that spanned fundamental software principles, test case scenarios, and the automation of these scenarios. This comprehensive examination not only gauged our theoretical understanding but also tested our practical application of QA concepts.

Feature: Job Title Management

Acceptance Criteria:

Acceptance Criteria 1: Adding a Job Title as an Administrator
*Given I am an administrator.
When I add a job title with the following details:
| Job Title | Job Description | Job Specification | Note
| [Title] | [Job Description] | [Job Specification] | [Note]
Then the job title should be successfully added*

Acceptance Criteria 2: Limiting Characters in Job Description Field
*Given I am adding a job title.
When I enter a job description with more than 400 characters
Then an error message should be displayed*

Acceptance Criteria 3: Limiting Size upload of Job Specification
*Given I am adding a job specification.
When I upload a job specification file exceeding 1 MB
Then an error message should be displayed*

Acceptance Criteria 4: Limiting Characters in Note Field
*Given I am adding a job title.
When I enter a note with more than 400 characters
Then an error message should be displayed*

Acceptance Criteria 5: Show error when clicking on cancel button with text in fields.
*Given I am adding a job title with entered text.
When I click the cancel button
Then an error message should be displayed*

Acceptance Criteria 6: Displaying Recently Created Jobs in the List

*Given there are existing job titles.
When I create new job titles
Then the new job titles should be displayed in the list*

Acceptance Criteria 7: Editing a Recently Created Job Title
*Given there is a recently created job title.
When I edit the job title with updated details
Then the changes should reflect when opening the job title*

Acceptance Criteria 8: Deleting a Job Title as an Administrator
*Given there is an existing job title.
When I delete the job title
Then the job title should be successfully deleted and not displayed in the list*

Acceptance Criteria 9: Restricting Deletion with Associated Employees
*Given there is an existing job title associated with employees.
When I attempt to delete the job title
Then an error should be displayed, and the job title should not be deleted.*

Figure 29, QA Testing Hackathon, Case Scenario

2. Daily Huddle & Progress Report

As part of our agile development process, I took on the responsibility of organizing a weekly meeting, known as the **daily huddle**, with other team members. This regular gathering provided a dedicated space for open communication and collaboration within the team. During these sessions, we discussed ongoing tasks, progress made, and any potential blockers hindering our workflow. The daily huddle served as a forum for team members to share insights, seek assistance, and collectively address challenges.

If any issues arose, we would collectively strategize and collaborate to overcome them. The Scrum Master played a pivotal role in facilitating these discussions, ensuring that everyone had the opportunity to voice concerns and that blockers were promptly addressed. This proactive approach not only enhanced our problem-solving capabilities but also strengthened team cohesion, fostering a supportive and collaborative environment geared towards achieving our project goals.

	C	D	E	F	G
1	Project Name	Project Description	Tasks	Issues raised	Remarks
2	Induction	Talks about the rules and core values at Sd Worx.	Health and Safety, Code of Ethics	None	Very informative.
3	Induction	About the services and benefits available at Sd Worx + Team ex	Health, Insurance, Car and Phone Loan Benefits + Teams Building exercises	None	We learnt a lot about the company.
4	Training	Angular Training through videos.	Watching an angular tutorial video.	None	None
5	Training	C# Training	Starting C# Tutorial: w3schools.com and Microsoft Learn	None	None
6	Training	C# Training, CRUD API Training	Continuing with C# Tutorial: w3schools.com and Microsoft Learn	None	None
7					
8	Training	C# and CRUD API Training	Tutorial: w3schools.com and Microsoft Learn, starting CRUD: concept of .NET 6 and	None	None
9	Training	CRUD API and Angular Training	Finishing tutorial on CRUD, starting Angular Tutorial: Tour of Heroes	SQL Connectivity Issues, Database migration issues	Solved
10	Training	Angular Training	Finishing Angular Tutorial: Tour of Heroes	Issues regarding the version of the in-memory Web API package f	Solved
11	Training	Angular Training through videos.	With Angular 14 & .NET 6 Web API, EF Core & SQL Server: Youtube Video to w	apiUrl issues, different versions of package installations	Managed to solve the issues.
12	Training	How-To's on project cloning, VS Navigation, VS Setup.	profile configuration. Package Managers tutorial (Yarn & NPM), Database setup, API setup and running.	None	None
13					
14	Training	SDEvents overview, Running tests	ebkit description, Webkit description and how to use it. running tests with 'npx pl	issues when using the path of a test file to run/debug that specific test	Managed to solve the issues.
15	Training	Playwright Tutorial	Watching tutorials on Playwright	Path of file issues	Managed to solve the issues.
16	Training	QA Training	ftware testing, Watching tutorials on Playwright, Training on Azure and Git-making	Issues on page/context fixtures in playwright	Managed to solve the issues.
17	Training	QA Training	Functional Requirements, Watching tutorials on Playwright, Training on Azure and C	None	None
18	SDEvents	Learning about the project, its requirements and its codes.	ulate database with existing data, Watching tutorials on Playwright, Code Walkthro	Unable to start the project using serve.	Had to use yarn start.
19					
20	SDEvents	Assigning specific tasks for SDEvents	scription, Implement Service to fetch Users from the database, Implement table th	None	None
21	SDEvents	Carrying out the tasks assigned	scription, Implement Service to fetch Users from the database, Implement table that displays a list of all user in the admin page etc.	Issues solved	Issues solved
22	SDEvents	Finishing out the tasks assigned + QA Training	scription, Implement Service to fetch Users from the database, Implement table th	the main navigational buttons, Issues when using dotnet ef for migrat	Issues solved
23	SDEvents	Corrected/ Reverted the changes according to the reviewers.	Reverting changes, according to reviewers+ re-committing	None	None
24	SDEvents	Improving methods.		None	None
25					
26	Training, SDEvents	QA Training	for both positive and negative scenarios, with one scenario per test case, Execute t	None	None
27	Training	QA Training	nd negative scenarios, with one scenario per test case, Execute the user story testin	None	None
28	Training	QA Training	nd negative scenarios, with one scenario per test case, Execute the user story testin	None	None
29	SDEvents	Corrected/ Reverted the changes according to the reviewers.	Corrected/ Reverted the changes according to the reviewers.	None	None
30	SDEvents	Corrected/ Reverted the changes according to the reviewers.	Corrected/ Reverted the changes according to the reviewers.	None	None

Figure 29.1, Daily Huddle 1

	C	D	E	F	G
25					
26	Training, SDEvents	QA Training	for both positive and negative scenarios, with one scenario per test case, Execute the user story testin	None	None
27	Training	QA Training	nd negative scenarios, with one scenario per test case, Execute the user story testin	None	None
28	Training	QA Training	nd negative scenarios, with one scenario per test case, Execute the user story testin	None	None
29	SDEvents	Corrected/ Reverted the changes according to the reviewers.	Corrected/ Reverted the changes according to the reviewers.	None	None
30	SDEvents	Corrected/ Reverted the changes according to the reviewers.	Corrected/ Reverted the changes according to the reviewers.	None	None
31					
32	SDEvents	SDEvents-calendar service	Issues related to calendar event services	EventCalendarService to Calendar Component, Resolution of iEver	Issues solved
33	SDEvents	Preparation for SDEvents-code review	To review and present a demo on our code	None	None
34	Training	QA Training	Recap on sprint cycle,Explanation on QA bug life cycle	None	None
35	Training	QA Training	User Story 3, User Story 4	None	None
36	SDEvents	Sprint Review and Training	QA exercises, User Story 5 and sprint review	None	None
37					
38	SDEvents	QA Exercises, Assigned project tasks	QA User Story Exercises and assigned tasks for SDEvents project	None	None
39	SDEvents	QA Exercises, Assigned project tasks	QA User Story Exercises and assigned tasks for SDEvents project	None	None
40	SDEvents	ual QA Exercises, Automation QA Exercises, Assigned project	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	VS Code not functioning properly	None
41	SDEvents	ual QA Exercises, Automation QA Exercises, Assigned project	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	///	None
42	SDEvents	Manual and Automation QA Exercises, Assigned project tasks	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	None	None
43					
44		Christmas			
45	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	Unable to load user's current data	None
46	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	Unable to load user's current data	None
47	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	Unable to load user's current data	None
48	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	A User Story Exercises (Manual & Automation) and assigned tasks for SDEvents proj	Unable to load user's current data	None
49					
50	///	New Year	///	///	///
51	///	New Year	///	///	///
52	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	QA - Automation using playwright	///	Setting Acquainted with Automation tool
53	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	QA - Automation using playwright	///	///
54	DEvents & QA Training	QA- Automation Testing, Assigned project tasks	QA - Automation using playwright, Code assessment	///	
55					

Figure 29.2, Daily Huddle 2

In addition to our weekly team meetings, I actively participated in personal progress reporting as a means of self-evaluation and reflection. Each week, I meticulously filled out a progress report, detailing the tasks I undertook, the progress achieved, and any noteworthy remarks or observations. This documentation served as a valuable tool for tracking my individual contributions to the project and provided a transparent overview of my weekly achievements.

The progress report also included insights gained from mentor guidance, acknowledging the valuable input and assistance provided by mentors throughout the week. Additionally, any remarks or challenges encountered during the week were highlighted, allowing for proactive problem-solving and continuous improvement. This regular practice of self-assessment not only facilitated effective communication with mentors but also contributed to my personal and professional growth by encouraging a reflective and goal-oriented mindset.

A	B
WEEKLY INTERNSHIP PROGRESS REPORT	
1	
3 Column1	Column2
4 Date	6-Nov-23
5 Project Name	Induction and Learning C#
6 Project Description	Understanding basic principles of C#
7 Trainees involved	Everyone
8 Tasks	Notes from w3schools.com and watching youtube tutorials
9 Coached by	Akash, Kishan
10 Remarks	Encouraged self learning
11	
12 Column1	Column2
13 Date	13-Nov-23
14 Project Name	Learning C# and Angular, Introduction to QA Testing
15 Project Description	CRUD with a .NET 6 Web API & Entity Framework Core and build a software application(Tour of heroes), Small introduction to Testing, CRUD with Angular 14 & .NET 6 (Web API, EF Core & SQL Server)
16 Trainees involved	Everyone
17 Tasks	Watch youtube tutorials and follow step by step guide on building the application
18 Coached by	Yogesh, Akash, Vashish, Shakile
19 Remarks	Enlightening and motivational
20	
21 Column1	Column2
22 Date	20-Nov-23
23 Project Name	Playwright tutorial, Git commands, Introduction to QA Testing, scrum and agile methodology.
24 Project Description	Learned automation testing, Used git commands (Fetch, Pull, Push), create branches.
25 Trainees involved	Everyone
26 Tasks	Watched playwright tutorial videos, Notes on Testing,
27 Coached by	Yogesh, Akash, Vashish, Shakile, Atish, Krisley, Kishan
28 Remarks	Motivational and learning on how to work as and in a team.
29	

Figure 30.1, Progress Report 1

A	B
WEEKLY INTERNSHIP PROGRESS REPORT	
30 Column1	Column2
31 Date	27-Nov-23
32 Project Name	QA Testing, Worked on task assigned by Mentors
33 Project Description	Worked on QA exercises and worked on project SDEvents(changed layout of events)
34 Trainees involved	Everyone
35 Tasks	QA Exercises and SDEvents
36 Coached by	Yogesh, Akash, Vashish, Kishan
37 Remarks	Experimental and growing responsibility
38	
39 Column1	Column2
40 Date	4-Dec-23
41 Project Name	QA Testing and Self Learning
42 Project Description	Worked on second set of exercises on QA Testing and user stories
43 Trainees involved	Everyone
44 Tasks	QA Exercises and SDEvents
45 Coached by	Yogesh, Akash, Kishan, Shakile, Vashish, Atish
46 Remarks	Experimental and practice
47	
48 Column1	Column2
49 Date	11-Dec-23
50 Project Name	QA Testing and SDEvents
51 Project Description	Worked on 3 User Story QA Exercises and task on SDEvent Project
52 Trainees involved	Everyone
53 Tasks	QA Exercises and SDEvents
54 Coached by	Yogesh, Akash
55 Remarks	Challenging and Experimental
56	

Figure 30.2, Progress Report 2

A	B
Column1	Column2
56 Date	18-Dec-23
57 Project Name	QA Testing and SDEvents
58 Project Description	Worked on user Story 3/4/5 QA Exercises and new task on SDEvent Project
59 Trainees involved	Everyone
60 Tasks	QA Exercises and adding a drawer to display events SDEvents
61 Coached by	Yogesh, Akash
62 Remarks	Inspirational and difficult
63	
Column1	Column2
64 Date	25-Dec-23
65 Project Name	SDEvents
66 Project Description	New Task on SDEvent
67 Trainees involved	Everyone
68 Tasks	Adding a drawer to display events info in the calendar
69 Coached by	Yogesh, Akash
70 Remarks	Challenging
71	
Column1	Column2
72 Date	1-Jan-24
73 Project Name	SDEvents
74 Project Description	New Task on SDEvent
75 Trainees involved	Everyone
76 Tasks	Adding a drawer to display events info in the calendar, preparing for hackathon
77 Coached by	Akash
78 Remarks	///
79	
Column1	Column2
80 Date	8-Jan-24
81 Project Name	SDEvents feature implementation, Dev Hackathon, QA Hackathon
82	

Figure 30.3, Progress Report 3

A	B
Column1	Column2
75 Date	1-Jan-24
76 Project Name	SDEvents
77 Project Description	New Task on SDEvent
78 Trainees involved	Everyone
79 Tasks	Adding a drawer to display events info in the calendar, preparing for hackathon
80 Coached by	Akash
81 Remarks	///
82	
Column1	Column2
83 Date	8-Jan-24
84 Project Name	SDEvents feature implementation, Dev Hackathon, QA Hackathon
85 Project Description	Implementation of events in SD Events Calendar
86 Trainees involved	Everyone
87 Tasks	Implementation of events in SD Events Calendar
88 Coached by	Akash, Yogesh
89 Remarks	///
90	

Figure 30.4, Progress Report 4

6.0 Personal Evaluation

During my internship at SDWorx, I can safely say that I gain valuable knowledge and skills in different areas of expertise which I would confidently use in the future with confidence.

- **Professional Growth**

Engaged in continuous learning and development, contributing to professional growth and acquired a deeper understanding of industry best practices and standards.

- **Adaptability**

Demonstrated adaptability by working with cutting-edge technologies and methodologies and successfully navigated through diverse tasks, showcasing versatility in skill application.

- **Collaborative Skills**

Exhibited effective collaboration by actively participating in regular team meetings and hackathons and worked cohesively with mentors, showcasing strong interpersonal and communication skills.

- **Problem-Solving Abilities**

Showcased problem-solving capabilities during the hackathon, addressing complex scenarios in quality assurance. Also applied critical thinking and analytical skills in resolving challenges encountered during project work.

- **Project Management**

Demonstrated project management skills by creating and maintaining a personal task tracker application and effectively handled tasks, timelines, and responsibilities within an Agile/Scrum framework.

- **Professional Conduct**

Maintained professionalism in the workplace through regular reporting, task tracking, and effective communication and consistently met project expectations and deadlines, reflecting a commitment to excellence.

- **Feedback Incorporation**

I actively sought feedback from mentors, illustrating a proactive approach to self-improvement and incorporated feedback into work, demonstrating a commitment to continuous improvement.

- **Commitment to Learning**

Displayed commitment to learning by participating in training sessions and applying acquired knowledge. Maintained a curiosity and eagerness to explore new technologies and methodologies.

7.0 Conclusion

In conclusion, my internship at SDWorx has been an enriching and transformative experience that has significantly contributed to both my personal and professional growth. Throughout the internship, I had the privilege of working on diverse projects that spanned software development and quality assurance, providing me with a holistic understanding of the industry's intricacies.

It has also instilled in me a sense of adaptability and readiness to embrace new challenges. The collaborative environment, regular team meetings, and hackathons underscored the importance of effective communication and teamwork, fostering a dynamic and supportive workplace culture. The mentorship provided valuable guidance and insights, and the emphasis on continuous learning has left an indelible mark on my approach to professional development.

As I conclude my internship, I carry forward a wealth of knowledge, skills, and a heightened enthusiasm for future endeavors in the dynamic field of software development and quality assurance. I am grateful for the opportunities, mentorship, and the supportive community at SDWorx, which have collectively shaped a transformative and memorable internship experience.

#sdinterns@work

Please join us in extending a warm welcome to our new interns! It is our last intake for 2023. We are thrilled to embark on this journey of growth, learning, and collaboration with these outstanding individuals.

As they immerse themselves in the vibrant culture of SD Worx, let's support and nurture their potential, knowing that their contributions will undoubtedly shape them as Professionals.

To our newest interns, welcome aboard! Your adventure with us begins now, and we can't wait to see the incredible impact you'll make.

Dharun, Abhisarika, Doorgesh, Ridhwaan, Kalimuddin, Manav, Yusra, Soubalakshmi, Khilesh, Esaie



Figure 31, Internship Team

8.0 Annex

1. Annex 1

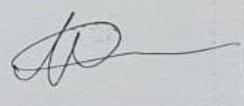
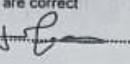
ICDT 2200 INDUSTRIAL TRAINING ATTENDANCE RECORD SHEET						Annex I
MONTH/YEAR:11/2023.- 12/2023.....						
	Date	Day	Time in	Time out	Total Hours Worked	Signature of Mentor
Week 1	06/11	Mon	09:00	17:00	8	
	07/11	Tues	09:00	17:00	8	
	08/11	Wed	09:00	17:00	8	
	09/11	Thur	09:00	17:00	8	
	10/11	Fri	09:00	17:00	8	
	11/11	Sat				
	12/11	Sun				
Week 2	13/11	Mon	09:00	17:00	8	
	14/11	Tues	09:00	17:00	8	
	15/11	Wed	09:00	17:00	8	
	16/11	Thur	09:00	17:00	8	
	17/11	Fri	09:00	17:00	8	
	18/11	Sat				
	19/11	Sun				
Week 3	20/11	Mon	09:00	17:00	8	
	21/11	Tues	09:00	17:00	8	
	22/11	Wed	09:00	17:00	8	
	23/11	Thur	09:00	17:00	8	
	24/11	Fri	09:00	17:00	8	
	25/11	Sat				
	26/11	Sun				
Week 4	27/11	Mon	09:00	17:00	8	
	28/11	Tues	09:00	17:00	8	
	29/11	Wed	09:00	17:00	8	
	30/11	Thur	09:00	17:00	8	
	01/12	Fri	09:00	17:00	8	
	02/12	Sat				
	03/12	Sun				
Week 5	04/12	Mon	09:00	17:00	8	
	05/12	Tues	09:00	17:00	8	
	06/12	Wed	09:00	17:00	8	
	07/12	Thur	09:00	17:00	8	
	08/12	Fri	09:00	17:00	8	
	09/12	Sat				
	10/12	Sun				
	No. of Days Absent:	0	FROM: 06/11	TO: 16/12	Total Hours Missed:	

Figure 32.1, Annex I – Attendance Sheet Pg1

	Total No. of Hours Worked:	200	
I certify that the details on this form are correct			
SIGNATURE OF STUDENT:			
	SIGNATURE OF MENTOR: 		

ATTENDANCE RECORD SHEET

MONTH/YEAR: 12/2023 – 01/2024

	Date	Day	Time in	Time out	Total Hours Worked	Signature of Mentor
Week 6	11/12	Mon	09:00	17:00	8	
	12/12	Tues	09:00	17:00	8	
	13/12	Wed	09:00	17:00	8	
	14/12	Thur	09:00	17:00	8	
	15/12	Fri	09:00	17:00	8	
	16/12	Sat				
	17/12	Sun				
Week 7	18/12	Mon	09:00	17:00	8	
	19/12	Tues	09:00	17:00	8	
	20/12	Wed	09:00	17:00	8	
	21/12	Thur	09:00	17:00	8	
	22/12	Fri	09:00	17:00	8	
	23/12	Sat				
	24/12	Sun				
Week 8	25/12	Mon	09:00	17:00	8	
	26/12	Tues	09:00	17:00	8	
	27/12	Wed	09:00	17:00	8	
	28/12	Thur	09:00	17:00	8	
	29/12	Fri	09:00	17:00	8	
	30/12	Sat				
	31/12	Sun				
Week 9	01/01	Mon	09:00	17:00	8	
	02/01	Tues	09:00	17:00	8	
	03/01	Wed	09:00	17:00	8	
	04/01	Thur	09:00	17:00	8	
	05/01	Fri	09:00	17:00	8	
	06/01	Sat				
	07/01	Sun				
Week 10	08/01	Mon	09:00	17:00	8	
	09/01	Tues	09:00	17:00	8	
	10/01	Wed	09:00	17:00	8	
	11/01	Thur	09:00	17:00	8	
	12/01	Fri	09:00	17:00	8	

Figure 32.2, Annex I - Attendance Sheet Pg2

		Sat					
		Sun					
	No. of Days Absent	0	FROM: 11/12	TO: 12/01	Total Hours Missed:		
	Total No. of Hours Worked:	200					
I certify that the details on this form are correct							
SIGNATURE OF STUDENT: <i>[Signature]</i>			SIGNATURE OF MENTOR: <i>[Signature]</i>				

Figure 32.3, Annex 1 – Attendance Sheet Pg3

Annex II

2. Annex II

ICDT 2200 INDUSTRIAL TRAINING

To be completed by the Academic Training Supervisor during the training

Visiting Academic Training Supervisor (ATS) Comments:

The ATS to comment on the student's progress following discussions with the student and the designated Mentor.

Comments:

The visiting ATS to record any advice given or learning action plans suggested to enhance the student's performance during this industrial training

Signature of ATS:..... Date:

ICDT 2200 INDUSTRIAL TRAINING**STUDENT REFLECTION****STUDENT'S PERSONAL REFLECTION ON THE INDUSTRIAL PLACEMENT****What knowledge, skills and personal attributes am I bringing to this placement?**

(These should include transferable knowledge, skills and personal attributes)

I acquired a versatile skill set that encompasses both development and quality assurance testing. I delved into programming languages such as C# and TypeScript, honing my proficiency in crafting efficient and scalable code. My exploration extended to web development, where I gained hands-on experience with HTML and Angular, mastering the art of building dynamic and responsive user interfaces. Simultaneously, my journey into quality assurance testing encompassed both manual and automation practices. I acquired a comprehensive understanding of manual testing methodologies, refining my ability to create robust test cases and ensure software reliability. Embracing automation, I familiarized myself with tools like Playwright, empowering me to automate test scenarios and contribute to the continuous improvement of software quality. This multifaceted learning experience has equipped me with a holistic skill set that allows me to navigate seamlessly between development and quality assurance.

Identify your own learning needs:(These should include specific knowledge or skills you want to gain/develop **other** than the stipulated learning outcomes for this placement which you have to achieve)

- **Advanced Frontend Frameworks:** While I've gained proficiency in Angular, I aim to explore and deepen my knowledge in other advanced frontend frameworks
- **Cloud Computing:** I aspire to enhance my understanding of cloud computing platforms, particularly AWS or Azure, to grasp the principles of deploying and managing applications in a cloud environment.
- **Advanced Testing Frameworks:** Building on my experience with Playwright, I am keen to explore and master other advanced testing frameworks such as Selenium or Cypress to broaden my skill set in test automation.
- **Agile Methodologies:** While I've gained exposure to Agile/Scrum during this placement, I aim to deepen my understanding of agile methodologies

- **Advanced Database Management:** I intend to delve deeper into advanced database management concepts, exploring optimization techniques, query performance tuning, and mastering database administration tasks.
- **Effective Communication.** While I have experience in communication through presentations and discussions, I aspire to refine my ability to convey complex technical concepts in a clear and concise manner. This includes improving my written communication skills for documentation and project reports.

Reflect on and record your personal achievements and any future development needs on completion of this industrial placement:

(The student's reflection should focus on the transferable knowledge, skills, attitudes and professional attributes gained and developed during this placement)

In terms of achievements, I can safely say that I received positive feedbacks from supervisors and other colleagues. I aspire to continue this behaviour of mine because it made me feel confident and able to make leaps of faith which resulted in good outcomes. My ability to adapt to diverse projects and tasks has been a key accomplishment, showcasing my flexibility in navigating different aspects of software development and quality assurance. Active participation in team collaborations, hackathons, and leadership roles has not only fostered strong interpersonal relationships but has also contributed to the establishment of a positive and supportive team culture. I've successfully balanced multiple responsibilities, demonstrating effective time management skills that ensured the timely completion of projects. Facing challenges during various tasks has honed my problem-solving skills, fostering a proactive mindset in seeking collaborative solutions.

Looking ahead, I aim to further refine my communication skills, strengthen my professional networking capabilities, and cultivate emotional intelligence for a more holistic and successful professional journey.

4. Annex IV

		Annex IV												
ICDT 2200 INDUSTRIAL TRAINING														
To be completed by the mentor at the end of the training														
<table border="1"><thead><tr><th></th><th style="text-align: center;">Yes</th><th style="text-align: center;">No</th></tr></thead><tbody><tr><td>The student has attended the training punctually</td><td style="text-align: center;"><input checked="" type="checkbox"/></td><td style="text-align: center;"><input type="checkbox"/></td></tr><tr><td>All evidences recorded in his/her portfolio reflect the factual learning achievement of the student during the training</td><td style="text-align: center;"><input checked="" type="checkbox"/></td><td style="text-align: center;"><input type="checkbox"/></td></tr><tr><td>The student has demonstrated satisfactory performance during the training</td><td style="text-align: center;"><input checked="" type="checkbox"/></td><td style="text-align: center;"><input type="checkbox"/></td></tr></tbody></table>				Yes	No	The student has attended the training punctually	<input checked="" type="checkbox"/>	<input type="checkbox"/>	All evidences recorded in his/her portfolio reflect the factual learning achievement of the student during the training	<input checked="" type="checkbox"/>	<input type="checkbox"/>	The student has demonstrated satisfactory performance during the training	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Yes	No												
The student has attended the training punctually	<input checked="" type="checkbox"/>	<input type="checkbox"/>												
All evidences recorded in his/her portfolio reflect the factual learning achievement of the student during the training	<input checked="" type="checkbox"/>	<input type="checkbox"/>												
The student has demonstrated satisfactory performance during the training	<input checked="" type="checkbox"/>	<input type="checkbox"/>												
Remarks Good performance overall. Attentive, punctual, fast-learning.														
.....														
Name of Mentor: Saval Yogenh Amaud														
Designation: Software Developer														
Signature of Mentor: 														
Date: 18/01/2024														
<p style="text-align: center;">Please insert rubber stamp of your organisation here</p> <div style="text-align: center; border: 1px solid black; padding: 5px; margin-top: 10px;"><p>SD Worx (Mauritius) Limited 11th Floor, NEX Tower, Rue du Savoir, Cybercity, 72201, Ebene</p></div>														

Figure 33, Annex IV

5. Offer of Placement form

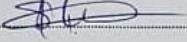
UNIVERSITY OF MAURITIUS FACULTY OF INFORMATION COMMUNICATION & DIGITAL TECHNOLOGIES OFFER OF PLACEMENT FORM (INDIVIDUAL)					
ORGANISATION	SD Worx LTD				
ADDRESS	Nex Tower, 11 th floor, Rue du Savoir Ebene				
TELEPHONE	407 1000	FAX		EMAIL	
CONTACT PERSON					
The Organisation agrees to offer an industrial placement to the student indicated below, for a period of ten weeks from Monday 16 October 2023 to 12 December 2023 (inclusive)					
Mr/Mrs:	Ronisha Pathen				
Signature:			Date:	15/11/2023	
Student Name:	Eraie Poos		Student Id: 0114769		
Programme of Study:	B.Sc. (Hons) Software Engineering Contact No: 58046654				
Please provide a brief description of expected tasks to be undertaken by the trainee(s)					
Mentor (On site Supervising Officer for the above trainee)					
Name	Forename		Designation (Team Leader, Analyst, Senior Programmer etc.)		
Mawah	Yogesh		Engineer/Developer Bias		
SD Worx (Mauritius) Limited 11 th Floor, NEX Tower, Rue du Savoir, Cybercity, 72201, Ebene					

Figure 34, Annex IV