Hearts Card Game

Secure Software Design and Programming
ISA/SWE 681

By

Aranta Rokade (G01049891)
Vaishnavi Konar (G01019190)
(Under the guidance of Dr. David A. Wheeler)

May 04, 2018

George Mason University
Fairfax, Virginia

Contents

1. INTRODUCTION 4
2. DESIGN AND ARCHITECTURE OF THE CODE 4
2.1. ARCHITECTURE 4
2.1.1. Components 4
2.1.1.1. Database-SQL Server 2016: 4
2.1.1.2. Hearts.DAL 5
2.1.1.3. Hearts.BAL 5
2.1.1.4. Hearts.MVC 5
2.1.1.5. Hearts.ViewModels 5
2.1.1.6. Utility 5
2.2. FUNCTIONALITY 5
2.2.1. Log in/Register/Logout 5
2.2.2. Join existing game or create a new game 6
2.2.3. Play the game 6
2.2.4. View Score Board 6
2.2.5. View Rules 6
3. INSTALLATION INSTRUCTIONS 7
4. OPERATING INSTRUCTIONS 7
5. GAME RULES 7
Players and Cards 7
Object of Game 7
Deal and Passing 8
The Play of the Hand 8
Scoring 8
6. WHY WE BELIEVE IT'S SECURE 9
6.1. Assurance Case 9
6.2. SSL/TLS 9
6.3. Security Misconfiguration 11
6.4. Cross Site Request Forgery (CSRF) 13
6.5. Cross Site Scripting(XSS) 15
6.5.1. ValidateInput attribute 15
6.5.2. RegularExpression attribute 15

6.5.3. Data Annotations 16
6.5.4. Autocomplete off 16
6.6. SQL Injection 17
6.7. Sensitive Data Exposure 17
6.7.1. Hashing 17
6.7.2. Encryption/ Decryption 18
6.7.3. Login Failure Details 18
6.8. Authentication and Session Management 19
6.9. Audit Trails and Logging 21
6.9.1. Audit Trails 21
6.9.2. Logging 21
6.10. Version Disclosure 23
6.11. Cache Control 25
6.12. Gameplay Rule Validations 26
6.13. OWASP ZAP Testing 29
6.13.1. Standard Mode – 04/24/2018 29
6.13.2. Attack Mode – 05/04/2018 29
7. Conclusion 30
REFERENCES 31

1. INTRODUCTION

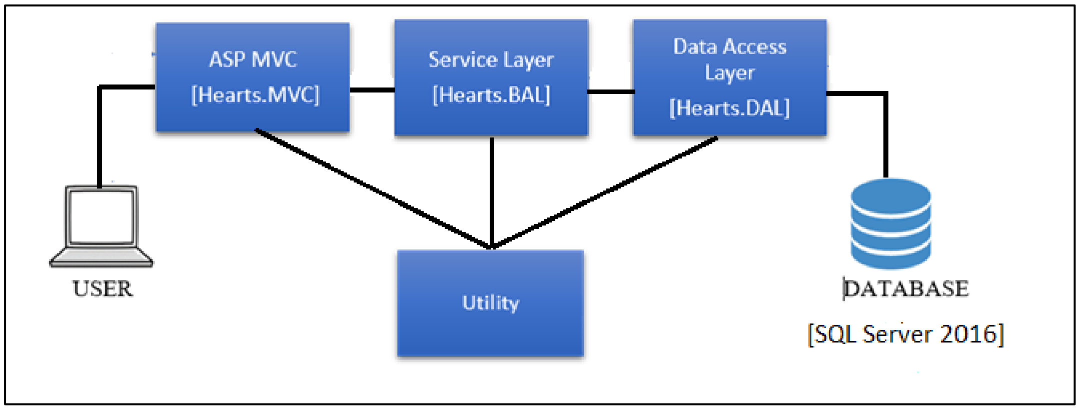
Hearts is a very easy game with many possible variations. The one created for our project uses pass the trash option and with four players. The objective of this game is to gain the lowest points possible while trying to eliminate all the cards in hand. The game ends when any one of the player reaches a total of 100 points. The player with the lowest number of points is declared as the winner.

2. DESIGN AND ARCHITECTURE OF THE CODE

This section involves the description of the major components that are a part of the games functionality.

2.1. ARCHITECTURE

Our game uses a client server architecture modeled using ASP.NET MVC architecture (.NET Framework) that separates the Model, View and Control of the Code. We have used .Net and Entity framework to develop the game. The image below shows architecture



2.1.1. Components

2.1.1.1. Database-SQL Server 2016:

We have created an instance of SQL server with Mixed Mode authentication, i.e., an application can access the database with username and password.

2.1.1.2. Hearts.DAL

This layer is the data access layer, used by the Hearts.BAL layer to access the database for CRUD [create/read/update/delete] operations. We have used Entity Framework 6.1.3 with Database First Approach. It allows the developer to use LINQ to Entity approach to save data to the database. This does not involve direct concatenation of SQL parameters and masks the underlying query.

2.1.1.3. Hearts.BAL

This layer contains the business logic of the application. The entire rules of the gameplay are developed in this layer. It is this layer that performs all the serve-side validation of the inputs and actions of the users. It is also responsible for passing data from the MVC to DAL layer and vice versa.

2.1.1.4. Hearts.MVC

This is the client of the game, modelled using MVC Framework. It contains the web pages which interact with the players. This layer also uses SignalR nuget package to send game play updates as push notifications to the players.

2.1.1.5. Hearts.ViewModels

This component contains the models used by Hearts.MVC project. For example, PlayerModel, GameModel, LoginModel, etc.

2.1.1.6. Utility

This component contains features which are accessed by all the other components. It includes the logic for Hashing, Encryption/Decryption, Logging and Custom Exception Handling.

2.2. FUNCTIONALITY

The functionalities provided in this application are listed below:

2.2.1. Log in/Register/Logout

A user can view certain features of the website without logging into the website, ex. rules, score board, etc. But, needs to create an account to play the game with other players. For this we have added registration, login and logout.

2.2.2. Join existing game or create a new game

A user can create a new game and wait for other players to join or join an existing game.

2.2.3. Play the game

This feature allows the user to play the game. A list of cards are dealt to the players and the website prompts the user to play when it's their turn.

2.2.4. View Score Board

This page contains the scores and the count of wins, losses and draws of all the players

2.2.5. View Rules

This game list all the rules incorporated in the gameplay.

3. INSTALLATION INSTRUCTIONS

1. Download Microsoft Visual Studio 2017.
2. Install SQL Server 2016.
3. Turn on IIS in Windows Features.
4. Create the game database using the DBScript.sql file.
5. Open the .sln file from the directory where the file is located.
6. Change the database connection string in the Web.config file based on the instance of the SQL server installed in machine.
7. Build the solution and publish the Hearts.MVC project to IIS Manager.
8. Run the website locally using the assigned port number.

4. OPERATING INSTRUCTIONS

1. Players must have an account or create one to play the game. Once they log in, they can join an existing game or create a new one.
2. The game starts when all four players join the game room. The game waits 90 secs for sufficient players to join the room, else the game is aborted.
3. Once the game has started, the players are dealt 13 cards each and asked to pass 3 cards as trash. The direction of the trash being passed depends on the round. After all the trashed cards are passed, the game begins.
4. An unauthenticated user can view the Home page, About, Rules page, Leader-Board page and Contact us page.
5. **The website allows multiple games to be proceed simultaneously, however, one player is restricted to only one game at a time.**

5. GAME RULES

Players and Cards

Hearts is most commonly played by 4 people. A standard 52 card deck is used without any trump suit. Each heart is worth one penalty point and the queen of spades is worth 13 penalty points. The other cards have no value.

Object of Game

The object is to avoid scoring points. The game is ended by someone reaching or going over 100 points, and the winner is the player with the lowest score at this point.

Deal and Passing

Deal and play are clockwise. All the cards are dealt out one at a time, so that everyone has 13.

On the first hand, after the deal, each player passes any three cards(except the 2 of Clubs) face-down to the player to their left. When passing cards, you must first select the cards to be passed and place them face-down, ready to be picked up by the receiving player; only then may you pick up the cards passed to you, look at them and add them to your hand.

On the second hand each player passes three cards to the player to their right, in the same way. On the third hand each player passes three cards to the player sitting opposite. On the fourth hand no cards are passed at all. The cycle then repeats until the end of the game.

The Play of the Hand

The person who holds the Two of clubs must lead it to the first trick. The other players, in clockwise order, must play a card of the suit which was led if possible. If they do not have a card of that suit, they may play any card. The person who played the highest card of the suit led wins the trick and leads the next trick.

It is illegal to lead a heart until after a heart has been played to a previous trick, unless your hand contains nothing but hearts. Discarding a heart, thus allowing hearts to be led in future, is called breaking hearts. In general, discarding a penalty card on a trick is called painting the trick.

A player whose hand consists entirely of hearts may lead any heart, thereby breaking hearts, even if hearts have not previously been broken.

Scoring

Normally, each player scores penalty points for cards in the tricks which they won. Each heart scores one point, and the queen of spades scores 13 points. However, if you manage to win all the scoring cards (which is known as a slam or shooting the moon), your score is reduced by 26 points, or you may choose instead to have all other players' scores increased by 26 points.

The game continues until one player has reached or exceeded 100 points at the end of a hand. The person with the lowest score is then the winner.

6. WHY WE BELIEVE IT'S SECURE

6.1. Assurance Case

The security of the application is demonstrated using an assurance given below:

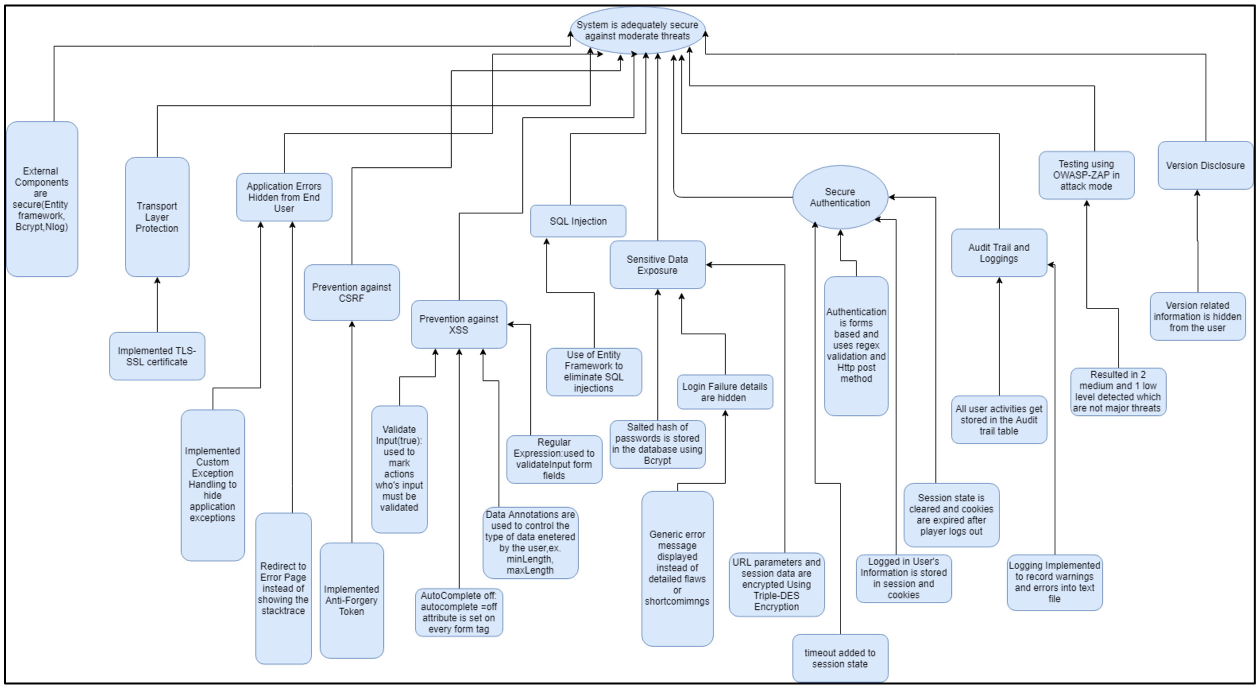
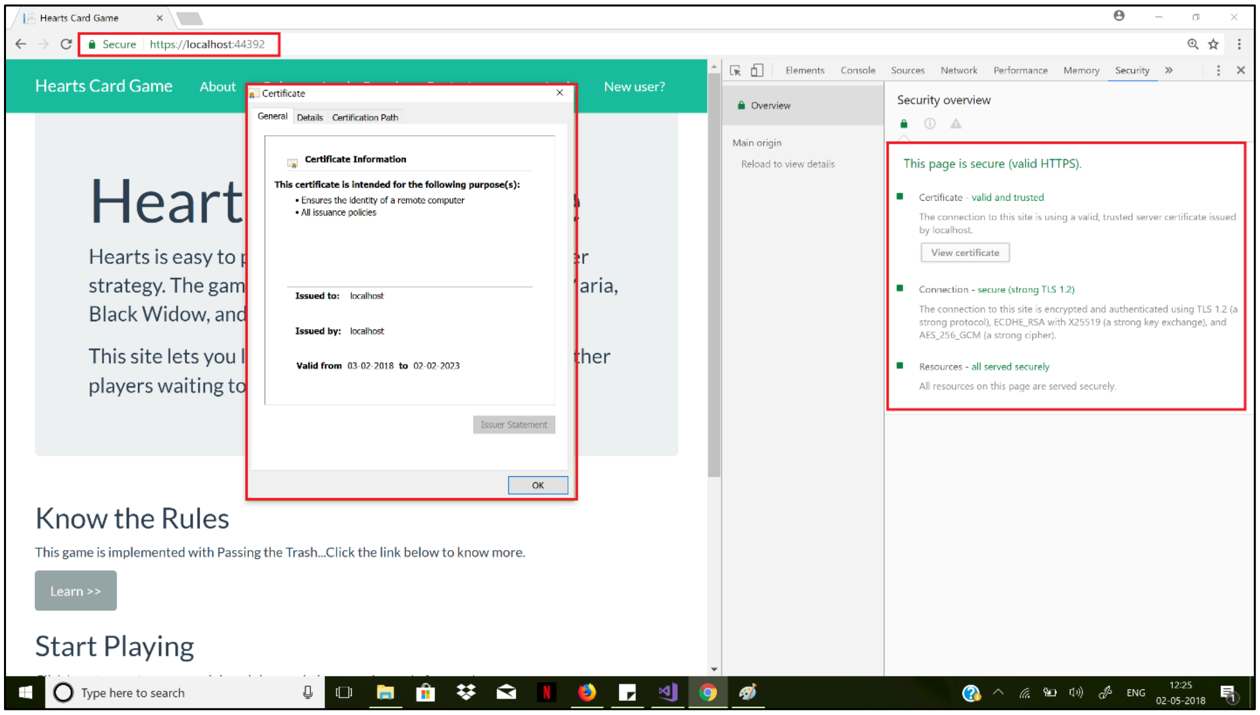

System is adequately secure against moderate threats
External Components are secure(Entity framework, Bcrypt,Nlog)
Transport Layer Protection
Implemented TLS-SSL certificate
Application Errors Hidden from End User
Implemented Custom Exception Handling to hide application exceptions
Redirect to Error Page instead of showing the stacktrace
Prevention against CSRF
Implemented Anti-Forgery Token
Prevention against XSS
Validate Input(true): used to mark actions who's input must be validated
AutoComplete off: autocomplete =off attribute is set on every form tag
Data Annotations are used to control the type of data enetered by the user,ex. minLength, maxLength
Regular Expression:used to validateInput form fields
SQL Injection
Use of Entity Framework to eliminate SQL injections
Sensitive Data Exposure
Salted hash of passwords is stored in the database using Bcrypt
Login Failure details are hidden
Generic error message displayed instead of detailed flaws or shortcomimngs
URL parameters and session data are encrypted Using Triple-DES Encryption
Secure Authentication
Authentication is forms based and uses regex validation and Http post method
timeout added to session state
Logged in User's Information is stored in session and cookies
Session state is cleared and cookies are expired after player logs out
Audit Trail and Loggings
All user activities get stored in the Audit trail table
Logging Implemented to record warnings and errors into text file
Testing using OWASP-ZAP in attack mode
Resulted in 2 medium and 1 low level detected which are not major threats
Version Disclosure
Version related information is hidden from the user

Fig. 6.1.1 Assurance Case Diagram.

6.2. SSL/TLS

"The TLS protocol aims primarily to provide privacy and data integrity between two communicating computer applications." [Dierks2008] This protocol helps keep the connection secure and the data transmitted encrypted. We have also generated a self-signed certificate using Microsoft Visual Studio and IIS Manager [WIKI-TLS]. The certificate uses RSA public key cryptography. We have added `[RequiredHttps]` attribute on all controllers to ensure SSL is enforced.

The images below show the certificate we created:



https://localhost:44392

Certificate — General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer
- All issuance policies

Issued to: localhost

Issued by: localhost

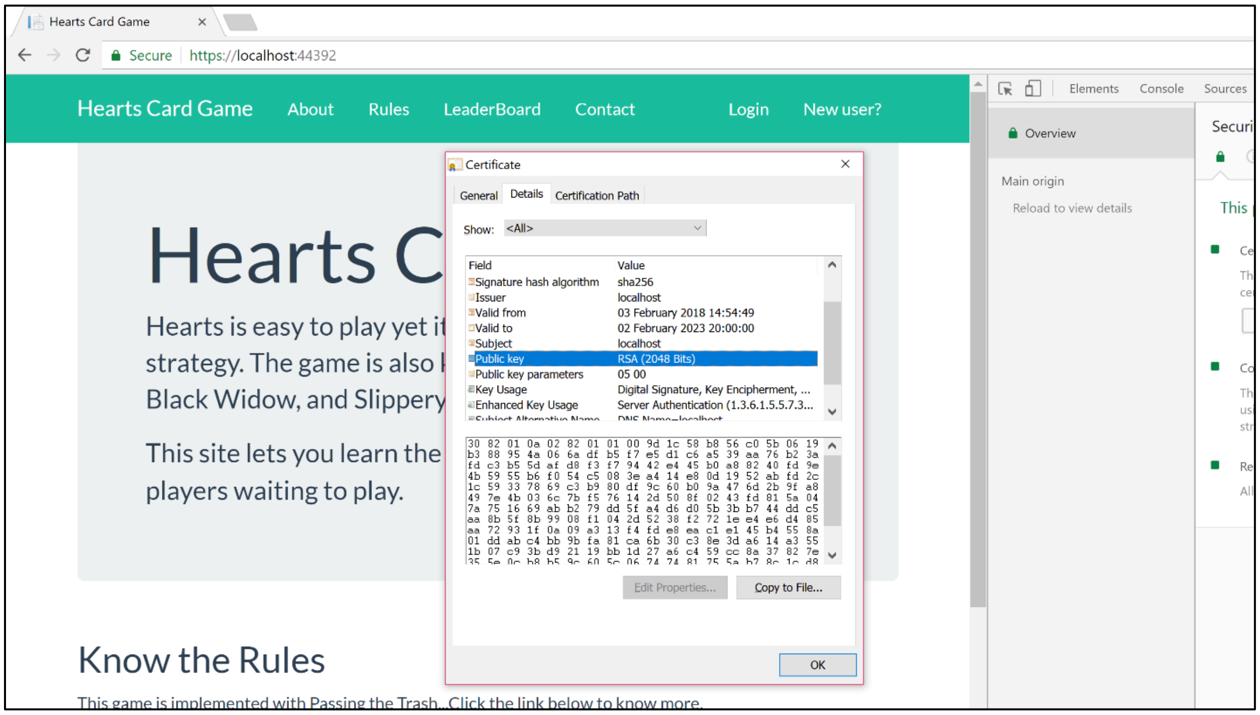
Valid from 03-02-2018 **to** 02-02-2023

Security overview

This page is secure (valid HTTPS).

- Certificate - valid and trusted
  The connection to this site is using a valid, trusted server certificate issued by localhost.
- Connection - secure (strong TLS 1.2)
  The connection to this site is encrypted and authenticated using TLS 1.2 (a strong protocol), ECDHE_RSA with X25519 (a strong key exchange), and AES_256_GCM (a strong cipher).
- Resources - all served securely
  All resources on this page are served securely.

Fig.6.2.1. Certificate



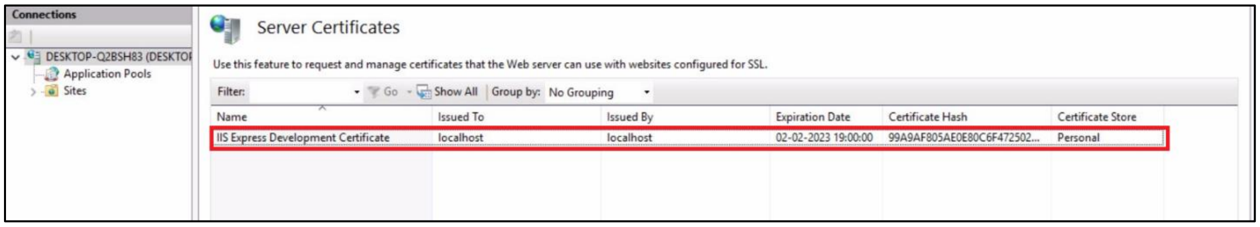
Certificate — General | Details | Certification Path

Show: <All>

Field	Value
Signature hash algorithm	sha256
Issuer	localhost
Valid from	03 February 2018 14:54:49
Valid to	02 February 2023 20:00:00
Subject	localhost
Public key	RSA (2048 Bits)
Public key parameters	05 00
Key Usage	Digital Signature, Key Encipherment, ...
Enhanced Key Usage	Server Authentication (1.3.6.1.5.5.7.3...

```
30 82 01 0a 02 82 01 01 00 9d 1c 58 b8 56 c0 5b 06 19
b3 88 95 4a 06 6a df b5 f7 e5 d1 c6 a5 39 aa 76 b2 3a
fd c3 b5 5d af d8 f3 f7 94 42 e4 45 b0 a8 82 40 fd 9e
4b 59 55 b6 f0 54 c5 08 3e a4 14 e8 0d 19 52 ab fd 2c
1c 59 33 78 69 c3 b9 80 df 9c 60 b0 9a 47 6d 2b 9f a8
49 7e 4b 03 6c 7b f5 76 14 2d 50 8f 02 43 fd 81 5a 04
7a 75 16 69 ab b2 79 dd 5f a4 d6 d0 5b 3b b7 44 dd c5
aa 8b 5f 8b 99 08 f1 04 2d 52 38 f2 72 1e e4 e6 d4 85
aa 72 93 1f 0a 09 a3 13 f4 fd e8 ea c1 e1 45 b4 55 8a
01 dd ab c4 bb 9b fa 81 ca 6b 30 c3 8e 3d a6 14 a3 55
1b 07 c9 3b d9 21 19 bb 1d 27 a6 c4 59 cc 8a 37 82 7e
```

Fig.6.2.2. Certificate Details



Server Certificates

Use this feature to request and manage certificates that the Web server can use with websites configured for SSL.

Name	Issued To	Issued By	Expiration Date	Certificate Hash	Certificate Store
IIS Express Development Certificate	localhost	localhost	02-02-2023 19:00:00	99A9AF805AE0E80C6F472502...	Personal

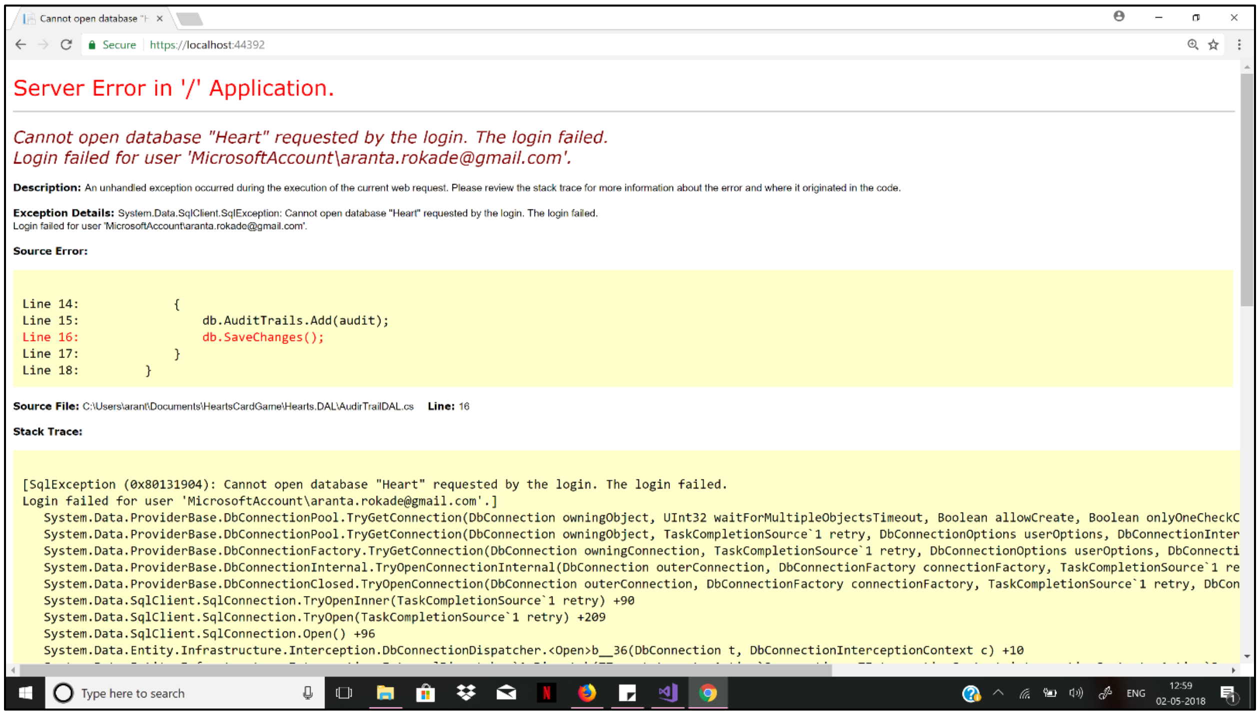
Fig.6.2.3. Certificate Hosted on IIS Server

6.3. Security Misconfiguration

Security Misconfiguration is an area where attackers attempt to gain unauthorized access to unprotected files and extra information. The use of default accounts or passwords, returning stack traces on errors, reveal information that can be exploited, this is caused due to improper Server or Web Application configuration. [OWASP-SEC-MISCONFIG]

In case of unforeseen circumstances, error messages display a stack trace, which usually looks like Fig.6.3.1. To handle this case gracefully, we created a custom error page and redirected the user to it, which doesn't reveal details about the error. After creating the error page, we add the following code to the Web.config file:

```
<customErrors mode="On" defaultRedirect="~/Error/Error.html" />
```



Server Error in '/' Application.

Cannot open database "Heart" requested by the login. The login failed.
Login failed for user 'MicrosoftAccount\aranta.rokade@gmail.com'.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Cannot open database "Heart" requested by the login. The login failed.
Login failed for user 'MicrosoftAccount\aranta.rokade@gmail.com'.

Source Error:

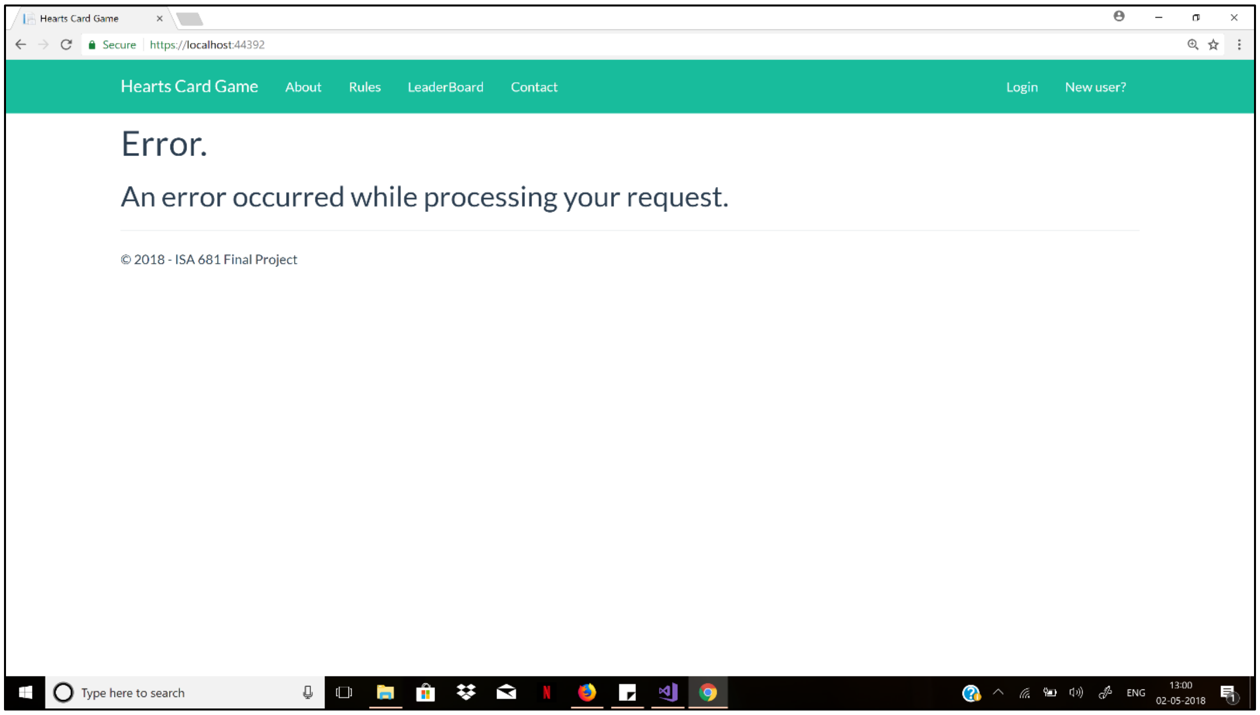
```
Line 14:             {
Line 15:                 db.AuditTrails.Add(audit);
Line 16:                 db.SaveChanges();
Line 17:             }
Line 18:         }
```

Source File: C:\Users\arant\Documents\HeartsCardGame\Hearts.DAL\AudirTrailDAL.cs **Line:** 16

Stack Trace:

```
[SqlException (0x80131904): Cannot open database "Heart" requested by the login. The login failed.
Login failed for user 'MicrosoftAccount\aranta.rokade@gmail.com'.]
   System.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, UInt32 waitForMultipleObjectsTimeout, Boolean allowCreate, Boolean onlyOneCheckC
   System.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, TaskCompletionSource`1 retry, DbConnectionOptions userOptions, DbConnectionInter
   System.Data.ProviderBase.DbConnectionFactory.TryGetConnection(DbConnection owningConnection, TaskCompletionSource`1 retry, DbConnectionOptions userOptions, DbConnecti
   System.Data.ProviderBase.DbConnectionInternal.TryOpenConnectionInternal(DbConnection outerConnection, DbConnectionFactory connectionFactory, TaskCompletionSource`1 re
   System.Data.ProviderBase.DbConnectionClosed.TryOpenConnection(DbConnection outerConnection, DbConnectionFactory connectionFactory, TaskCompletionSource`1 retry, DbCon
   System.Data.SqlClient.SqlConnection.TryOpenInner(TaskCompletionSource`1 retry) +90
   System.Data.SqlClient.SqlConnection.TryOpen(TaskCompletionSource`1 retry) +209
   System.Data.SqlClient.SqlConnection.Open() +96
   System.Data.Entity.Infrastructure.Interception.DbConnectionDispatcher.<Open>b__36(DbConnection t, DbConnectionInterceptionContext c) +10
```

Fig. 6.3.1 Stack Trace without Error Handling.



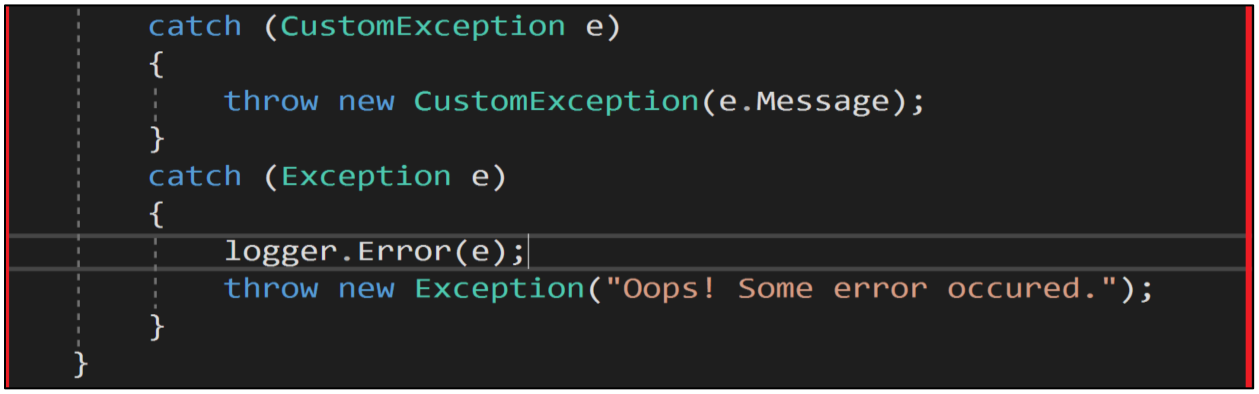
Error.

An error occurred while processing your request.

© 2018 - ISA 681 Final Project

Fig.6.3.2. Custom Error Page

When a user tries to make invalid moves in a game or accesses a wrong game, the application raises errors to warn the user. To differentiate between application generated exceptions and system generated exceptions we have created a wrapper around the `Exception` class called `CustomException`. If the `catch` clause catches a `CustomException`, it means it was thrown by the application deliberately to show the error to the end user, thus, can be forwarded as it is. Whereas, if it is not a `CustomException`, it means that it was an unknown exception and needs to be masked from the end user by modifying to a message such, “Oops! Some error occurred.”.



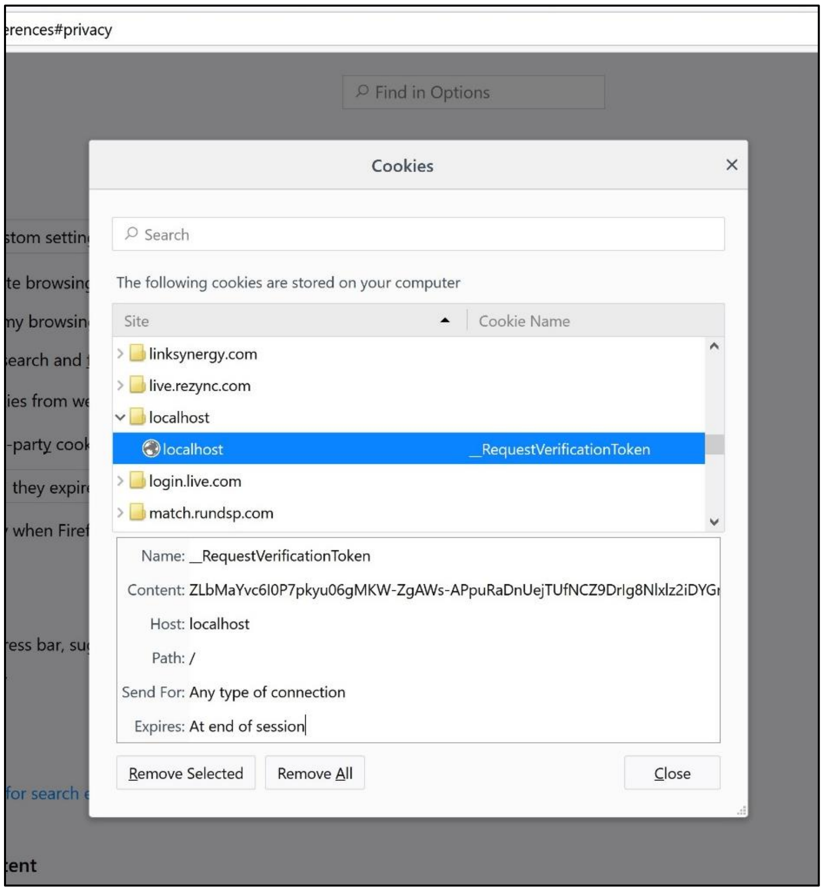
```
        catch (CustomException e)
        {
            throw new CustomException(e.Message);
        }
        catch (Exception e)
        {
            logger.Error(e);
            throw new Exception("Oops! Some error occured.");
        }
    }
```

Fig.6.3.2.

6.4. Cross Site Request Forgery (CSRF)

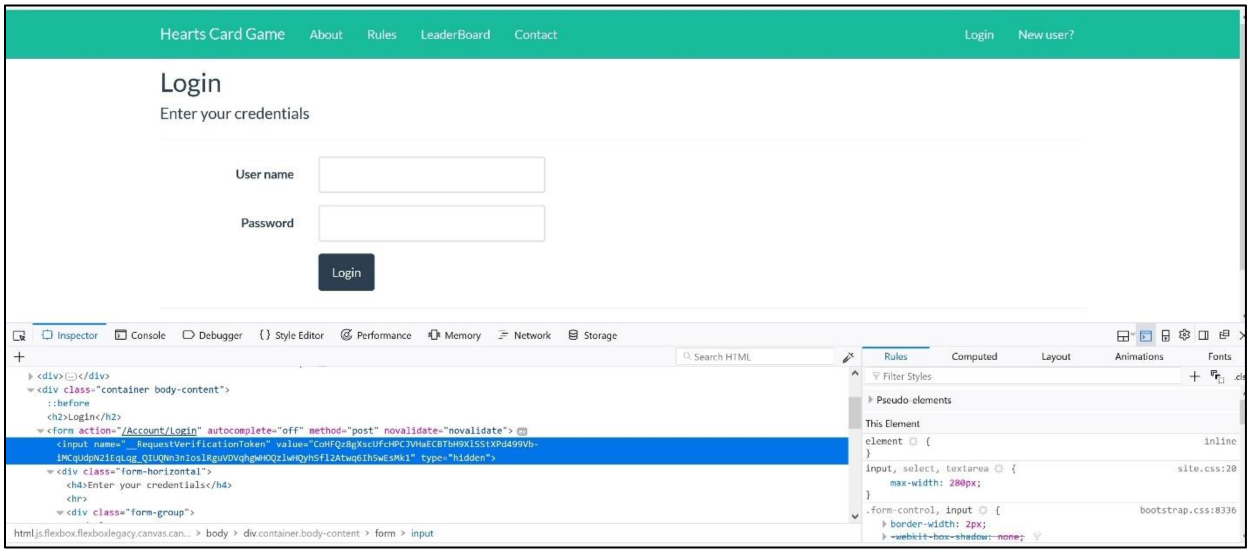
When a user is logged into the account, on a web application, an attacker can trick them into executing commands to perform state changing requests. An attack on a user with administrative rights may compromise the application. [OWASP-CSRF]

To overcome this, Microsoft has provided an AntiForgeryToken. To implement this, we added `@Html.AntiForgeryToken()` helper inside the form tag and `[ValidateAntiForgeryToken]` attribute which will check if the token is valid. Fig. 6.4.1 and 6.4.2 show the token data generated using AntiForgeryToken.



Name: __RequestVerificationToken
Content: ZLbMaYvc6I0P7pkyu06gMKW-ZgAWs-APpuRaDnUejTUfNCZ9DrIg8NlxIz2iDYGr
Host: localhost
Path: /
Send For: Any type of connection
Expires: At end of session

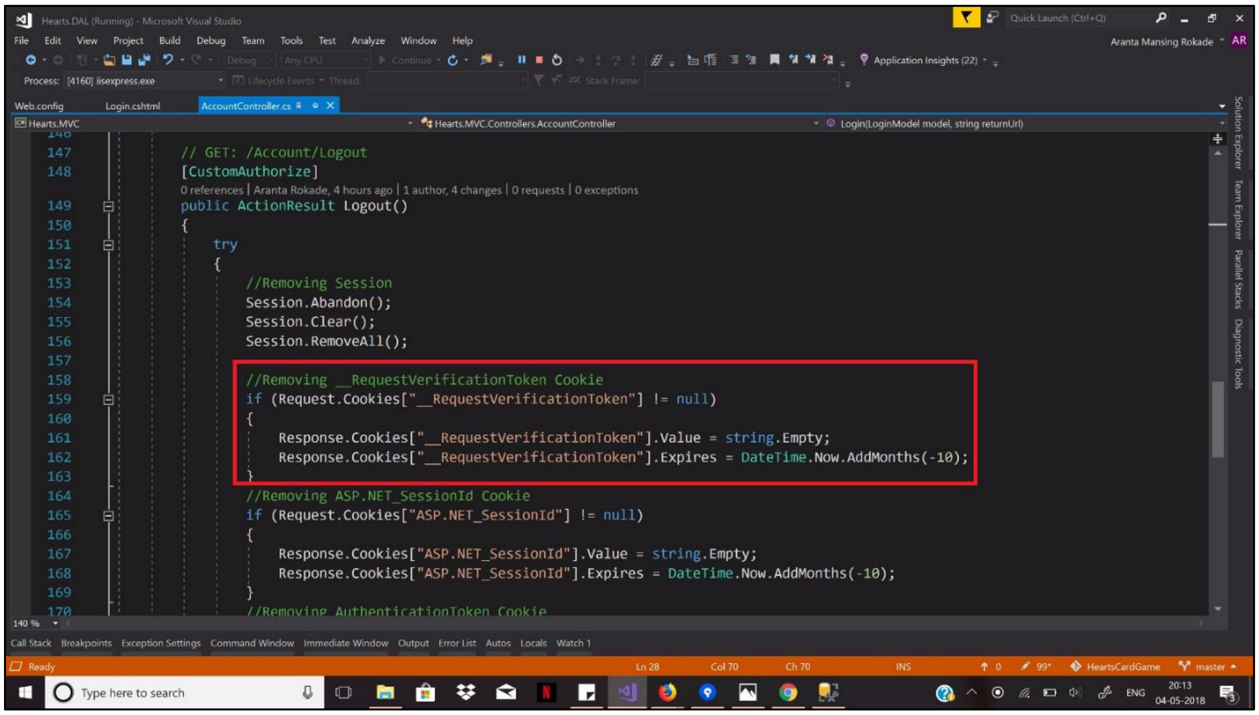
Fig.6.4.1. RequestVerificationToken cookie created



```
<div>…</div>
<div class="container body-content">
  ::before
  <h2>Login</h2>
  <form action="/Account/Login" autocomplete="off" method="post" novalidate="novalidate">
    <input name="__RequestVerificationToken" value="CoHFQz8gXscUfcHPCJVHaECBTbH9X1SStXPd499Vb-
    1MCqUdpN2iEqLqg_QIUQNn3nIoslRguVDVqhgWHOQzlwHQyhSfl2Atwq6Ih5wEsMk1" type="hidden">
    <div class="form-horizontal">
      <h4>Enter your credentials</h4>
      <hr>
      <div class="form-group">
```

Fig.6.4.2. RequestVerificationToken hidden element in form.

We also made sure that the tokens are completely removed on logout.


Hearts.DAL (Running) - Microsoft Visual Studio
Web.config
Login.cshtml
AccountController.cs
Hearts.MVC
Hearts.MVC.Controllers.AccountController
Login(LoginModel model, string returnUrl)

```
        // GET: /Account/Logout
        [CustomAuthorize]
        0 references | Aranta Rokade, 4 hours ago | 1 author, 4 changes | 0 requests | 0 exceptions
        public ActionResult Logout()
        {
            try
            {
                //Removing Session
                Session.Abandon();
                Session.Clear();
                Session.RemoveAll();

                //Removing __RequestVerificationToken Cookie
                if (Request.Cookies["__RequestVerificationToken"] != null)
                {
                    Response.Cookies["__RequestVerificationToken"].Value = string.Empty;
                    Response.Cookies["__RequestVerificationToken"].Expires = DateTime.Now.AddMonths(-10);
                }
                //Removing ASP.NET_SessionId Cookie
                if (Request.Cookies["ASP.NET_SessionId"] != null)
                {
                    Response.Cookies["ASP.NET_SessionId"].Value = string.Empty;
                    Response.Cookies["ASP.NET_SessionId"].Expires = DateTime.Now.AddMonths(-10);
                }
                //Removing AuthenticationToken Cookie
```

Fig.6.4.3. Code to clear RequestVerificationToken cookie after logout

6.5. Cross Site Scripting(XSS)

“Cross-site Scripting (XSS) is an attack in which malicious scripts are injected via input fields this attack is most common and allows an attacker to steal credentials and valuable data that can lead to a big security breach.” [Bageri2016]

The malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page. [OWASP-XSS]

To prevent our application from XSS attacks we have implemented the following measures:

6.5.1. ValidateInput attribute

The `[ValidateInput(true)]` attribute is an attribute that is used to mark action methods whose input must be validated. We have also used `ModelState.isValid()` function check to whether the model received by the action methods is valid.

6.5.2. RegularExpression attribute

We have added [RegularExpression] attribute to validate the username and password fields on the RegistrationModel, so that the user creates secure credentials.

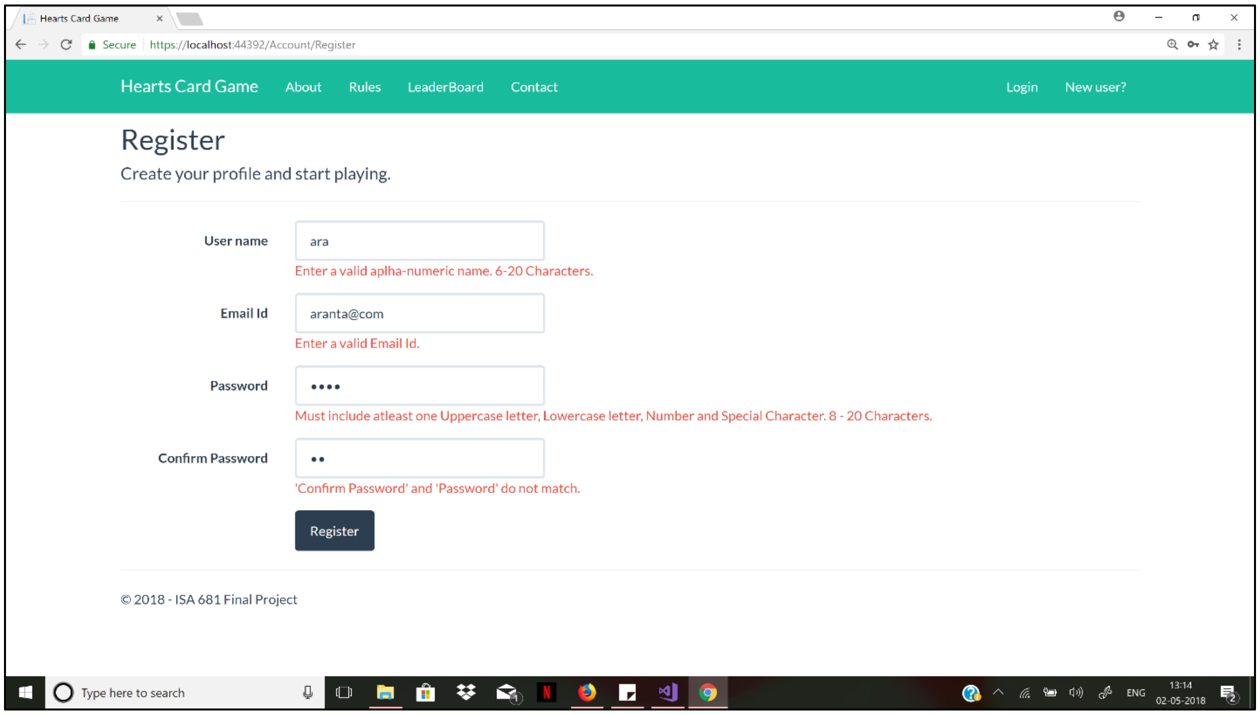
“Some regexes can take potential time and memory to process certain data. “[Wheeler2017]. A timer is set for 5 seconds to execute a single matching operation, this will prevent a user from exhausting the server by entering a malicious string. This attribute is not used on the Login Model, to reduce the details displayed to the user. The regex is evaluated on the server and hence the validation cannot be tampered with. We have created the regex using whitelisting mechanism.

Example 1:

```
[RegularExpression("^[a-zA-Z0-9]{6,20}$",
ErrorMessage = "Enter a valid aplha-numeric name. 6-20 Characters.",
MatchTimeoutInMilliseconds = 5)]
public string UserName { get; set; }
```

Example2:

```
[RegularExpression(@"^(?=.[a-z])(?=.[A-Z])(?=.\d)(?=.[^\da-zA-Z]).{8,20}$",
ErrorMessage = "Must include atleast one Uppercase letter, Lowercase letter, Number and Special Character. 8 - 20 Characters.",
MatchTimeoutInMilliseconds = 5)]
public string Password { get; set; }
```



Register

Create your profile and start playing.

User name	ara
	Enter a valid aplha-numeric name. 6-20 Characters.
Email Id	aranta@com
	Enter a valid Email Id.
Password	••••
	Must include atleast one Uppercase letter, Lowercase letter, Number and Special Character. 8 - 20 Characters.
Confirm Password	••
	'Confirm Password' and 'Password' do not match.

Fig. 6.5.1. Validation using regex and data annotations

6.5.3. Data Annotations

Data Annotations are used to control the type of data entered by the user. Below are a few examples used in our application.

`[Required]`: Used to specify a field as required.

`[EmailAddress(ErrorMessage = "Enter a valid Email Id.")]`: Internally uses regex validation for email addresses.

`[DataType(DataType.Password)]`: Prevents shoulder surfing by hiding the password

`[Compare("Password")]`: Matches the password and confirm password fields.

6.5.4. Autocomplete off

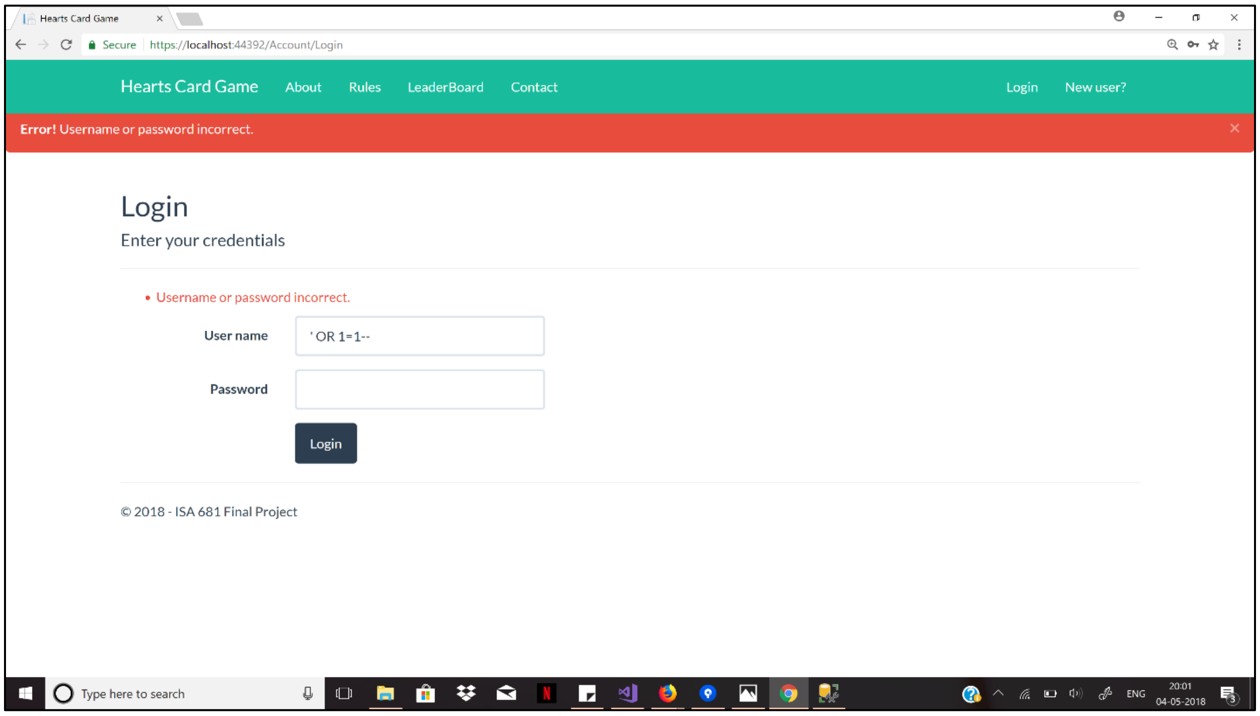
Although most browsers ignore the `autocomplete` attribute, we have still set its value to `off`.

The application does not have fields for file inputs or HTML based inputs. Thus, reduces the risks involved with multipart data.

6.6. SQL Injection

“A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application.” [OWASP-SQL]. This can be used to execute administrative operations on the database and can reveal valuable data to the attacker.

Entity framework uses ORM, Object relational Mapper, which is used to map SQL object to domain object in C#. Using Entity Framework eliminates SQL Injection attacks, as it internally converts LINQ to Entity queries to parameterized queries. Use of the Entity Framework is a very effective SQL injection prevention mechanism. [OWASP-DOTNET]



Error! Username or password incorrect.

Login

Enter your credentials

- Username or password incorrect.

User name: ' OR 1=1--

Password:

Fig.6.6.1. SQL injection Prevented

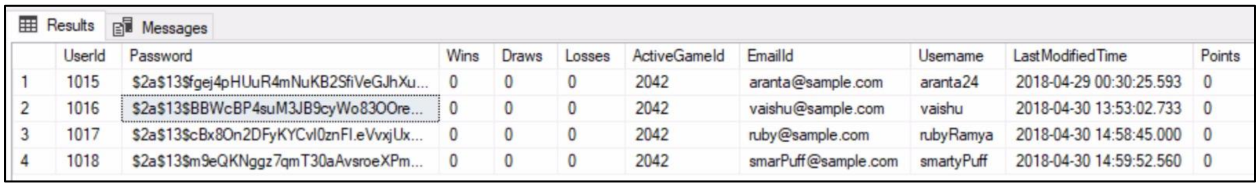
6.7. Sensitive Data Exposure

To avoid sensitive data from being exposed we have taken the following measures:

6.7.1. Hashing

Passwords are stored and validated as salted hashes in the database. We have used Bcrypt.Net nuget package to implement hashing.

```
public string Hash(string word)
{
     return BCrypt.Net.BCrypt.HashPassword(word, workFactor);
}

public bool ValidateHash(string word, string hash)
{
     return BCrypt.Net.BCrypt.Verify(word, hash);
}
```



Results
Messages

	UserId	Password	Wins	Draws	Losses	ActiveGameId	EmailId	Username	LastModifiedTime	Points
1	1015	$2a$13$fgej4pHUuR4mNuKB2SfiVeGJhXu...	0	0	0	2042	aranta@sample.com	aranta24	2018-04-29 00:30:25.593	0
2	1016	$2a$13$BBWcBP4suM3JB9cyWo83OOre...	0	0	0	2042	vaishu@sample.com	vaishu	2018-04-30 13:53:02.733	0
3	1017	$2a$13$cBx8On2DFyKYCvl0znFl.eVvxjUx...	0	0	0	2042	ruby@sample.com	rubyRamya	2018-04-30 14:58:45.000	0
4	1018	$2a$13$m9eQKNggz7qmT30aAvsroeXPm...	0	0	0	2042	smarPuff@sample.com	smartyPuff	2018-04-30 14:59:52.560	0

Fig. 6.7.1.1. Database with hashed passwords

6.7.2. Encryption/ Decryption

The URL parameters are encoded using Triple DES Encryption which masks the gameID from being exposed. Below is an image of how our URL looks.

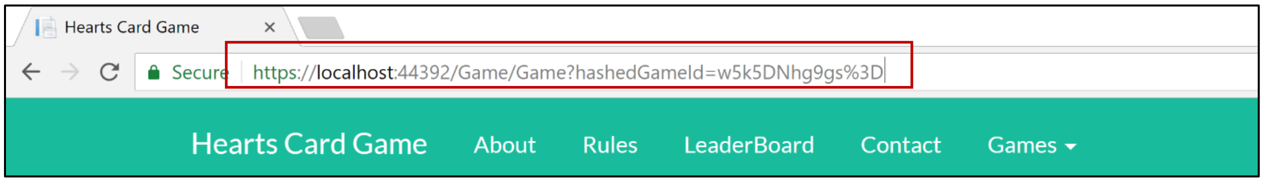
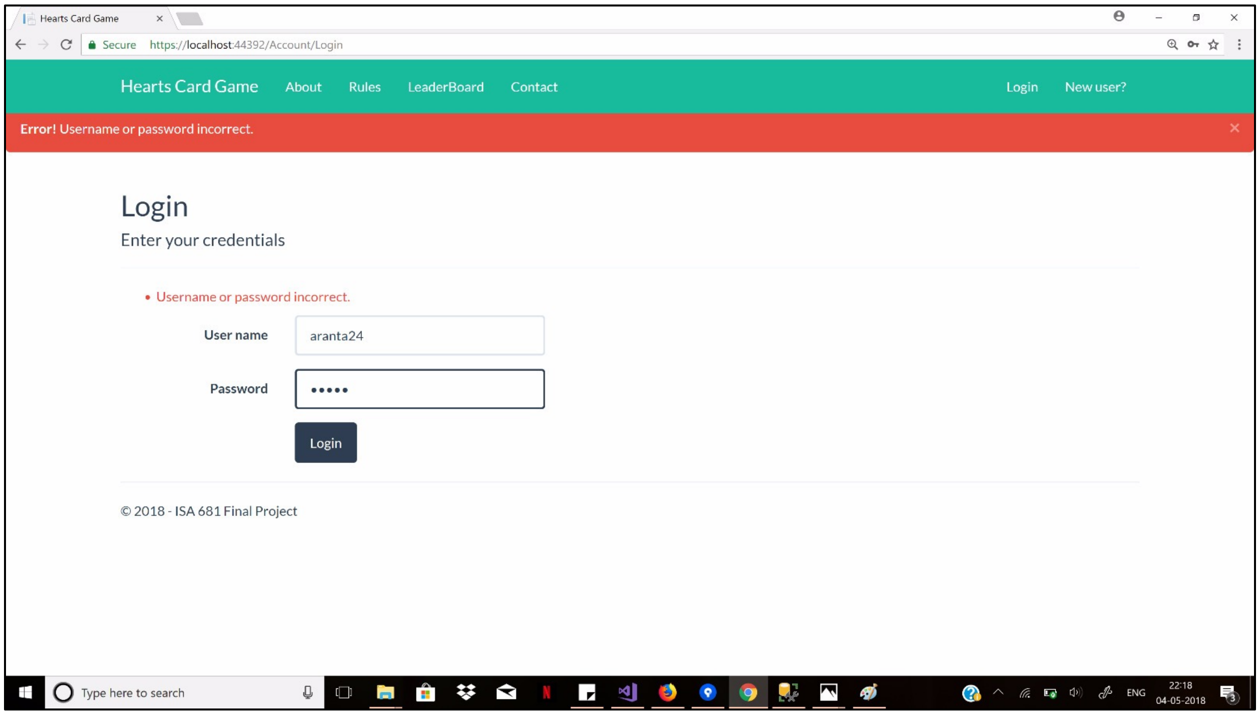


Fig.6.7.2.1 Encrypted URL

6.7.3. Login Failure Details

When a user types an incorrect username and password, the application doesn't reveal the shortcomings of the credentials, to prevent attackers from figuring out valid usernames.[Wheeler2018]


Error! Username or password incorrect.

Login

Enter your credentials

• Username or password incorrect.

User name: aranta24

Password: •••••

© 2018 - ISA 681 Final Project

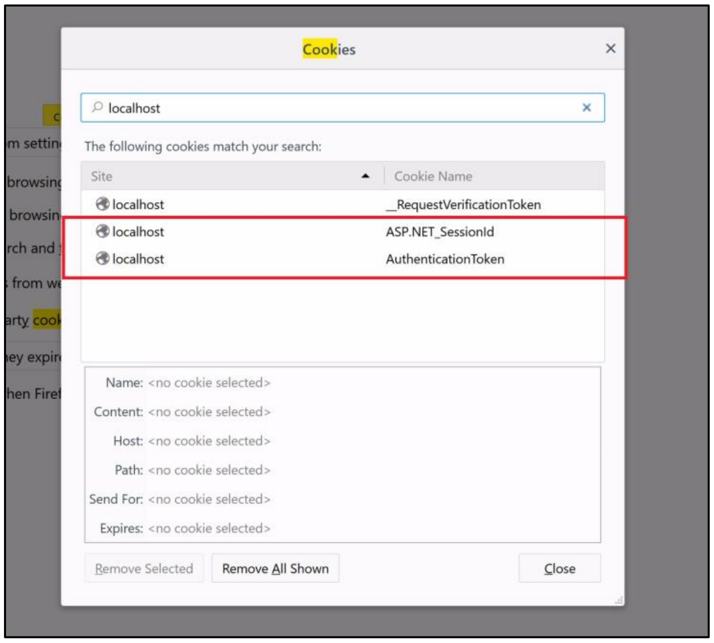
Fig.6.7.3. Minimal details revealed.

6.8. Authentication and Session Management

Authentication in this application is Forms-Based, i.e. username and password. After a user logs into the application, a **Session** gets created and the data is stored in **Cookies**.

We have ensured to clear the session data and expire the cookies after a user logs out. The data in the session is also encrypted and stored using Triple DES encryption. A timeout is added on the session using the tags given below in the Web.config file:

```
<sessionState timeout="180"></sessionState>

<httpCookies requireSSL="true" />
```



Cookies

localhost

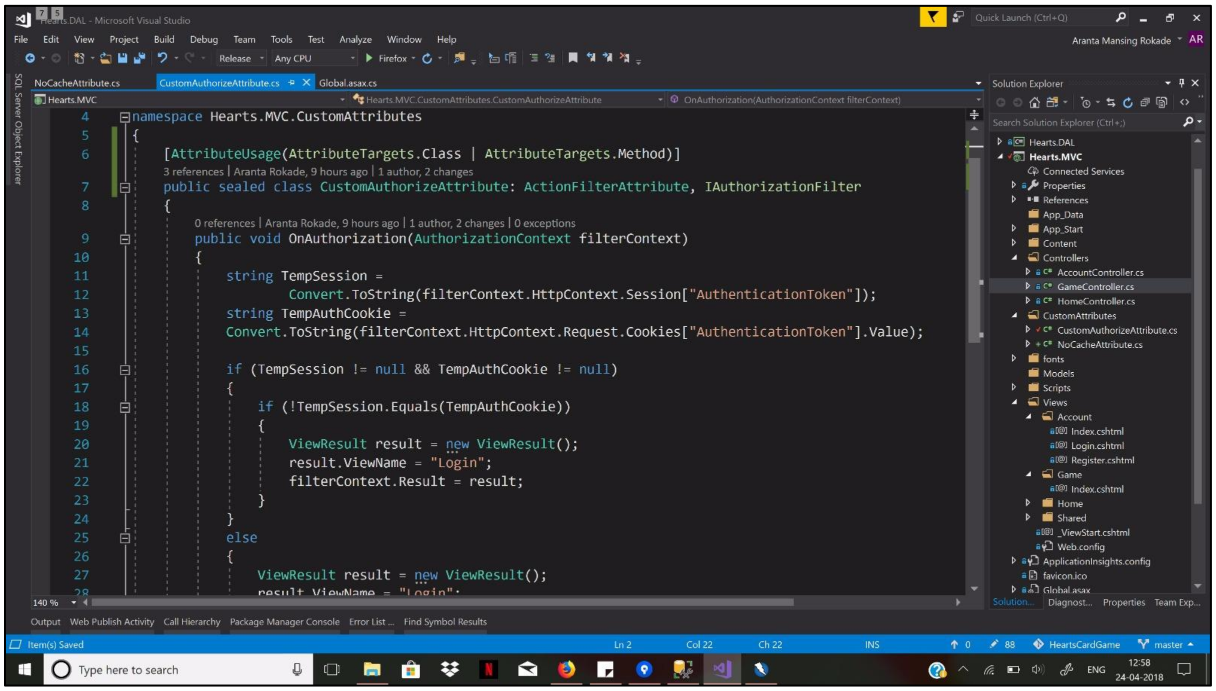
The following cookies match your search:

Site	Cookie Name
localhost	__RequestVerificationToken
localhost	ASP.NET_SessionId
localhost	AuthenticationToken

Name: <no cookie selected>
Content: <no cookie selected>
Host: <no cookie selected>
Path: <no cookie selected>
Send For: <no cookie selected>
Expires: <no cookie selected>

Fig. 6.8.1. Cookie Data

We have created an attribute to handle unauthorized users. This attribute is added on all the actions and controllers which require the user to be logged in. If an unauthorized user tries to access such a page, the `[CustomAuthorize]` attribute will redirect the him/her to the login page.



```
namespace Hearts.MVC.CustomAttributes
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method)]
    3 references | Aranta Rokade, 9 hours ago | 1 author, 2 changes
    public sealed class CustomAuthorizeAttribute: ActionFilterAttribute, IAuthorizationFilter
    {
        0 references | Aranta Rokade, 9 hours ago | 1 author, 2 changes | 0 exceptions
        public void OnAuthorization(AuthorizationContext filterContext)
        {
            string TempSession =
                    Convert.ToString(filterContext.HttpContext.Session["AuthenticationToken"]);
            string TempAuthCookie =
            Convert.ToString(filterContext.HttpContext.Request.Cookies["AuthenticationToken"].Value);

            if (TempSession != null && TempAuthCookie != null)
            {
                if (!TempSession.Equals(TempAuthCookie))
                {
                    ViewResult result = new ViewResult();
                    result.ViewName = "Login";
                    filterContext.Result = result;
                }
            }
            else
            {
                ViewResult result = new ViewResult();
                result.ViewName = "Login";
```

Fig. 6.8.2 CustomAuthorize attribute

6.9. Audit Trails and Logging

Audit trails and logs are used to keep track of user activity. “This aids in debugging, intrusion detection, etc.” [Wheeler-Lec8]

6.9.1. Audit Trails

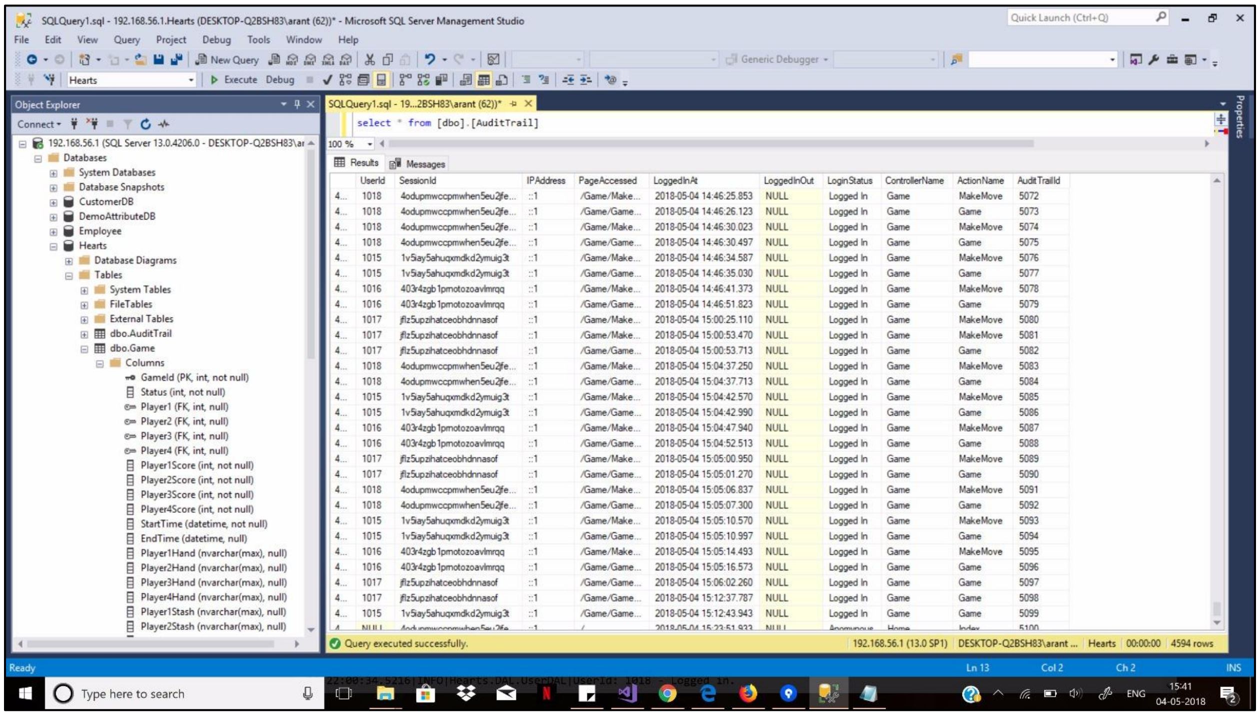
Our database consists of an audit table which logs all user activity. Every time a user accesses an action method from the controller, data, such as IP Address, accessed page, controller name, action name and, userID and login details if the user is logged in, gets stored in the database table. We have implemented this by creating an `[AuditTrailFilter]` which gets invoked every time a call to an action method is made. See Fig. 6.9.1.1.

6.9.2. Logging

We have also added logging to a text file to log warnings, information, errors, etc. generated by the application. We have implemented this by using the NLog nuget package. This package allowed us to easily log the data using simple code snippets such as:

```
logger.Debug("Sample debug message");
logger.Info("Sample informational message");
logger.Warn("Sample warning message");
logger.Error("Sample error message");
```

The logs generated by the application can be seen in Fig.6.9.2.1



SQLQuery1.sql - 192.168.56.1.Hearts (DESKTOP-Q2BSH83\arant (62))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
Object Explorer
SQLQuery1.sql - 19...2BSH83\arant (62))*
Results
Messages

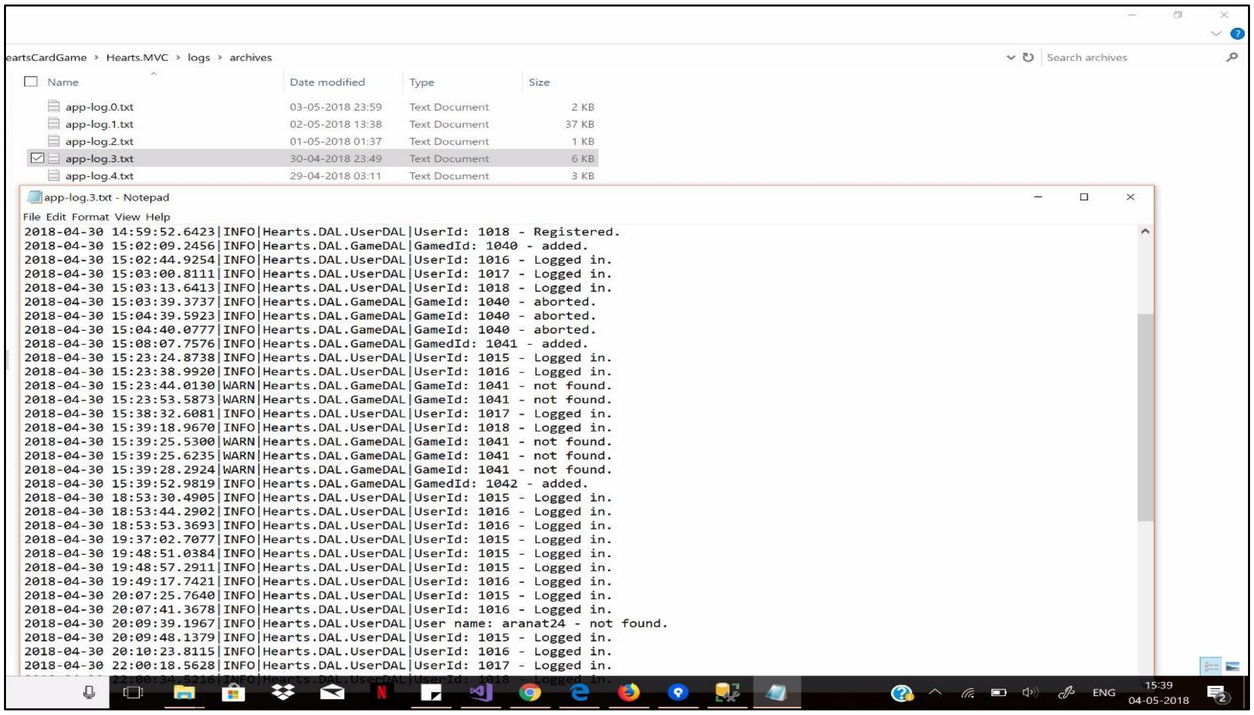
```
select * from [dbo].[AuditTrail]
```

	UserId	SessionId	IPAddress	PageAccessed	LoggedInAt	LoggedInOut	LoginStatus	ControllerName	ActionName	AuditTrailId
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Make...	2018-05-04 14:46:25.853	NULL	Logged In	Game	MakeMove	5072
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Game...	2018-05-04 14:46:26.123	NULL	Logged In	Game	Game	5073
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Make...	2018-05-04 14:46:30.023	NULL	Logged In	Game	MakeMove	5074
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Game...	2018-05-04 14:46:30.497	NULL	Logged In	Game	Game	5075
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Make...	2018-05-04 14:46:34.587	NULL	Logged In	Game	MakeMove	5076
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Game...	2018-05-04 14:46:35.030	NULL	Logged In	Game	Game	5077
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Make...	2018-05-04 14:46:41.373	NULL	Logged In	Game	MakeMove	5078
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Game...	2018-05-04 14:46:51.823	NULL	Logged In	Game	Game	5079
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Make...	2018-05-04 15:00:25.110	NULL	Logged In	Game	MakeMove	5080
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Make...	2018-05-04 15:00:53.470	NULL	Logged In	Game	MakeMove	5081
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Game...	2018-05-04 15:00:53.713	NULL	Logged In	Game	Game	5082
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Make...	2018-05-04 15:04:37.250	NULL	Logged In	Game	MakeMove	5083
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Game...	2018-05-04 15:04:37.713	NULL	Logged In	Game	Game	5084
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Make...	2018-05-04 15:04:42.570	NULL	Logged In	Game	MakeMove	5085
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Game...	2018-05-04 15:04:42.990	NULL	Logged In	Game	Game	5086
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Make...	2018-05-04 15:04:47.940	NULL	Logged In	Game	MakeMove	5087
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Game...	2018-05-04 15:04:52.513	NULL	Logged In	Game	Game	5088
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Make...	2018-05-04 15:05:00.950	NULL	Logged In	Game	MakeMove	5089
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Game...	2018-05-04 15:05:01.270	NULL	Logged In	Game	Game	5090
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Make...	2018-05-04 15:05:06.837	NULL	Logged In	Game	MakeMove	5091
4...	1018	4odupmwccpmwhen5eu2jfe...	::1	/Game/Game...	2018-05-04 15:05:07.300	NULL	Logged In	Game	Game	5092
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Make...	2018-05-04 15:05:10.570	NULL	Logged In	Game	MakeMove	5093
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Game...	2018-05-04 15:05:10.997	NULL	Logged In	Game	Game	5094
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Make...	2018-05-04 15:05:14.493	NULL	Logged In	Game	MakeMove	5095
4...	1016	403r4zgb1pmotozoavlmrqq	::1	/Game/Game...	2018-05-04 15:05:16.573	NULL	Logged In	Game	Game	5096
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Game...	2018-05-04 15:06:02.260	NULL	Logged In	Game	Game	5097
4...	1017	jflz5upzihatceobhdnnasof	::1	/Game/Game...	2018-05-04 15:12:37.787	NULL	Logged In	Game	Game	5098
4...	1015	1v5iay5ahuqxmdkd2ymuig3t	::1	/Game/Game...	2018-05-04 15:12:43.943	NULL	Logged In	Game	Game	5099
4...	NULL	4odupmwccpmwhen5eu2jfe...	::1	/	2018-05-04 15:23:51.933	NULL	Anonymous	Home	Index	5100

```
Query executed successfully.    192.168.56.1 (13.0 SP1)    DESKTOP-Q2BSH83\arant ...    Hearts    00:00:00    4594 rows
```

```
22:00:34.5216|INFO|Hearts.DAL.UserDAL|UserId: 1018 - Logged in.
```

Fig. 6.9.1.1 AuditTrails



eartsCardGame › Hearts.MVC › logs › archives

Name	Date modified	Type	Size
app-log.0.txt	03-05-2018 23:59	Text Document	2 KB
app-log.1.txt	02-05-2018 13:38	Text Document	37 KB
app-log.2.txt	01-05-2018 01:37	Text Document	1 KB
app-log.3.txt	30-04-2018 23:49	Text Document	6 KB
app-log.4.txt	29-04-2018 03:11	Text Document	3 KB

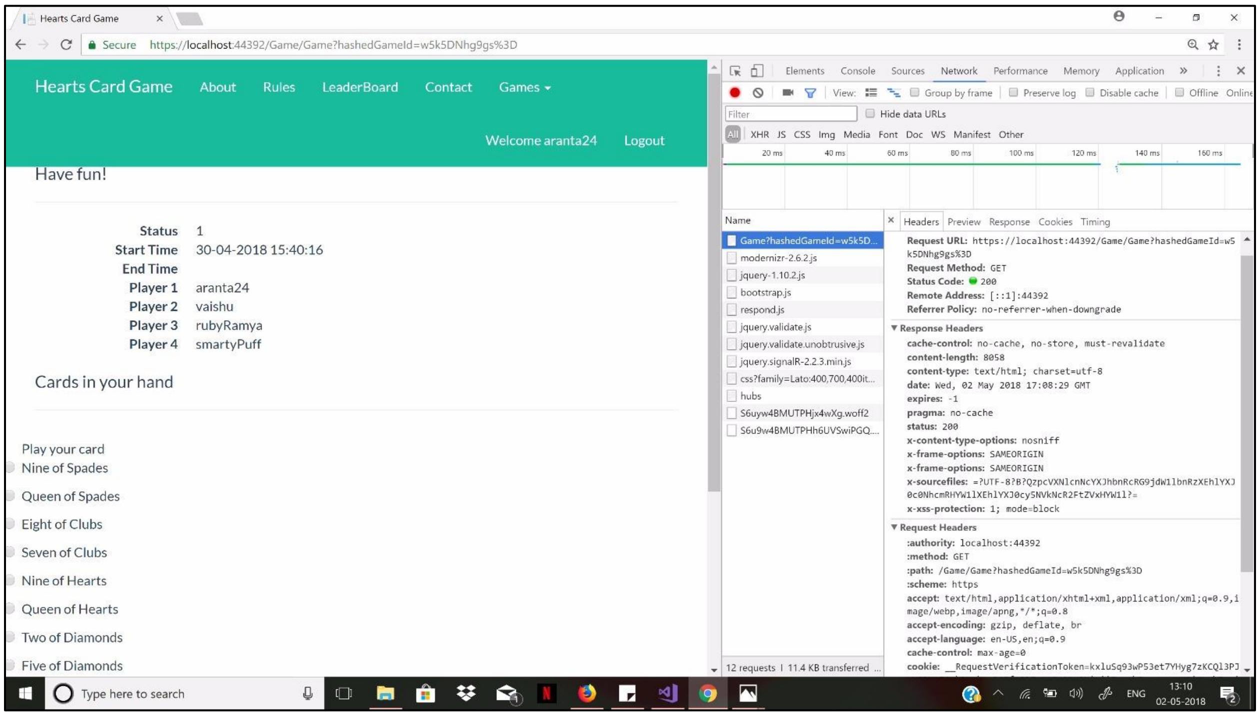
app-log.3.txt - Notepad

```
2018-04-30 14:59:52.6423|INFO|Hearts.DAL.UserDAL|UserId: 1018 - Registered.
2018-04-30 15:02:09.2456|INFO|Hearts.DAL.GameDAL|GamedId: 1040 - added.
2018-04-30 15:02:44.9254|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 15:03:00.8111|INFO|Hearts.DAL.UserDAL|UserId: 1017 - Logged in.
2018-04-30 15:03:13.6413|INFO|Hearts.DAL.UserDAL|UserId: 1018 - Logged in.
2018-04-30 15:03:39.3737|INFO|Hearts.DAL.GameDAL|GameId: 1040 - aborted.
2018-04-30 15:04:39.5923|INFO|Hearts.DAL.GameDAL|GameId: 1040 - aborted.
2018-04-30 15:04:40.0777|INFO|Hearts.DAL.GameDAL|GameId: 1040 - aborted.
2018-04-30 15:08:07.7576|INFO|Hearts.DAL.GameDAL|GamedId: 1041 - added.
2018-04-30 15:23:24.8738|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 15:23:38.9920|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 15:23:44.0130|WARN|Hearts.DAL.GameDAL|GameId: 1041 - not found.
2018-04-30 15:23:53.5873|WARN|Hearts.DAL.GameDAL|GameId: 1041 - not found.
2018-04-30 15:38:32.6081|INFO|Hearts.DAL.UserDAL|UserId: 1017 - Logged in.
2018-04-30 15:39:18.9670|INFO|Hearts.DAL.UserDAL|UserId: 1018 - Logged in.
2018-04-30 15:39:25.5300|WARN|Hearts.DAL.GameDAL|GameId: 1041 - not found.
2018-04-30 15:39:25.6235|WARN|Hearts.DAL.GameDAL|GameId: 1041 - not found.
2018-04-30 15:39:28.2924|WARN|Hearts.DAL.GameDAL|GameId: 1041 - not found.
2018-04-30 15:39:52.9819|INFO|Hearts.DAL.GameDAL|GamedId: 1042 - added.
2018-04-30 18:53:30.4905|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 18:53:44.2902|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 18:53:53.3693|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 19:37:02.7077|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 19:48:51.0384|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 19:48:57.2911|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 19:49:17.7421|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 20:07:25.7640|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 20:07:41.3678|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 20:09:39.1967|INFO|Hearts.DAL.UserDAL|User name: aranat24 - not found.
2018-04-30 20:09:48.1379|INFO|Hearts.DAL.UserDAL|UserId: 1015 - Logged in.
2018-04-30 20:10:23.8115|INFO|Hearts.DAL.UserDAL|UserId: 1016 - Logged in.
2018-04-30 22:00:18.5628|INFO|Hearts.DAL.UserDAL|UserId: 1017 - Logged in.
[illegible] 22:00:34.5216|INFO|Hearts.DAL.UserDAL|UserId: 1018 - Logged in.
```

Fig. 6.9.2.1. Log files

6.10. Version Disclosure

Version information can be used by an attacker to target specific attacks based on the vulnerabilities of the version. Whenever a browser sends request to the server, we get response headers which contain information related to the server. We have configured our application to remove such response headers and hide this data.



Have fun!

```
    Status  1
Start Time  30-04-2018 15:40:16
  End Time
  Player 1  aranta24
  Player 2  vaishu
  Player 3  rubyRamya
  Player 4  smartyPuff
```

Cards in your hand

Play your card
Nine of Spades
Queen of Spades
Eight of Clubs
Seven of Clubs
Nine of Hearts
Queen of Hearts
Two of Diamonds
Five of Diamonds

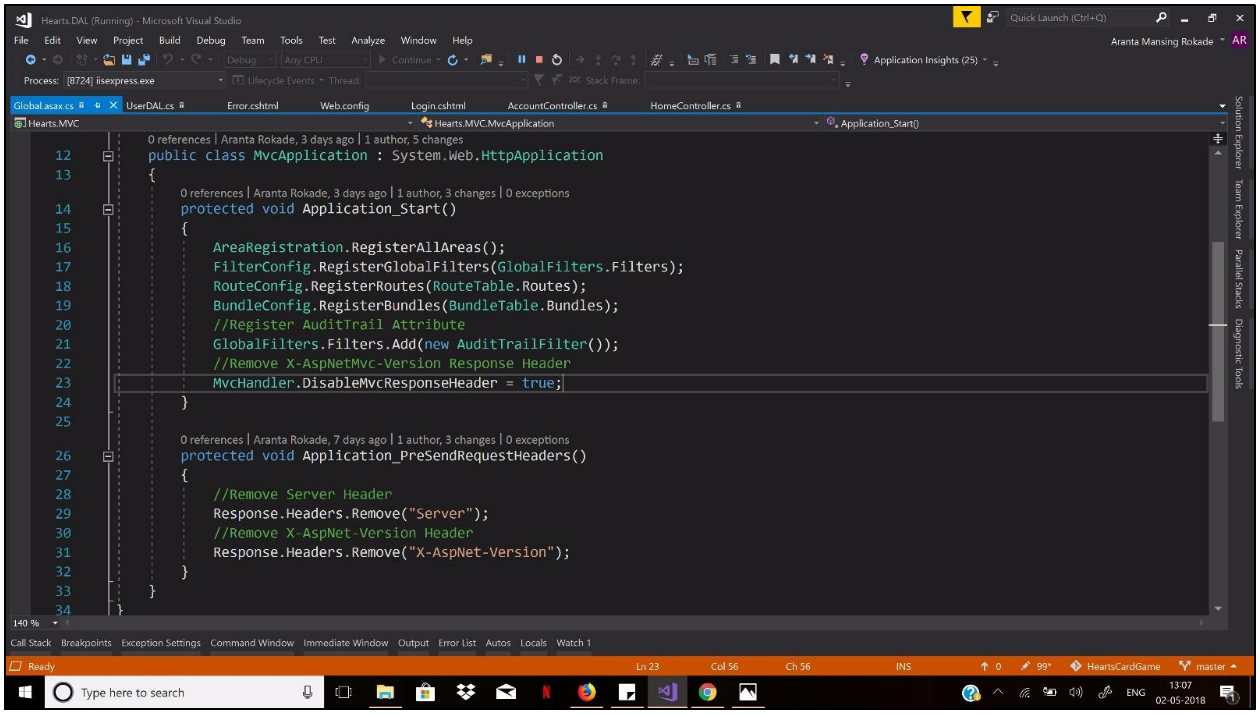
```
Request URL: https://localhost:44392/Game/Game?hashedGameId=w5k5DNhg9gs%3D
Request Method: GET
Status Code: 200
Remote Address: [::1]:44392
Referrer Policy: no-referrer-when-downgrade
▼ Response Headers
cache-control: no-cache, no-store, must-revalidate
content-length: 8058
content-type: text/html; charset=utf-8
date: Wed, 02 May 2018 17:08:29 GMT
expires: -1
pragma: no-cache
status: 200
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-frame-options: SAMEORIGIN
x-sourcefiles: =?UTF-8?B?QzpcVXNlcnNcYXJhbnRcRG9jdW1lbnRzXEhlYXJ0c0NhcmRHYW1lXEhlYXJ0cy5NVkNcR2FtZVxHYW1l?=
x-xss-protection: 1; mode=block
▼ Request Headers
:authority: localhost:44392
:method: GET
:path: /Game/Game?hashedGameId=w5k5DNhg9gs%3D
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
accept-encoding: gzip, deflate, br
accept-language: en-US,en;q=0.9
cache-control: max-age=0
cookie: __RequestVerificationToken=kxluSq93wP53et7YHyg7zKCQl3PJ
```

Fig. 6.10.1. Http Response Headers

We have added the following code in the Web.config file to set the appropriate response headers:

```
<httpProtocol>
  <customHeaders>
     <remove name="X-Powered-By" />
     <add name="X-Content-Type-Options" value="nosniff" />
     <add name="X-XSS-Protection" value="1; mode=block" />
     <add name="X-Frame-Options" value="SAMEORIGIN" />
  </customHeaders>
</httpProtocol>
```

Some of the headers were set in the application start file as shown in Fig 6.10.1

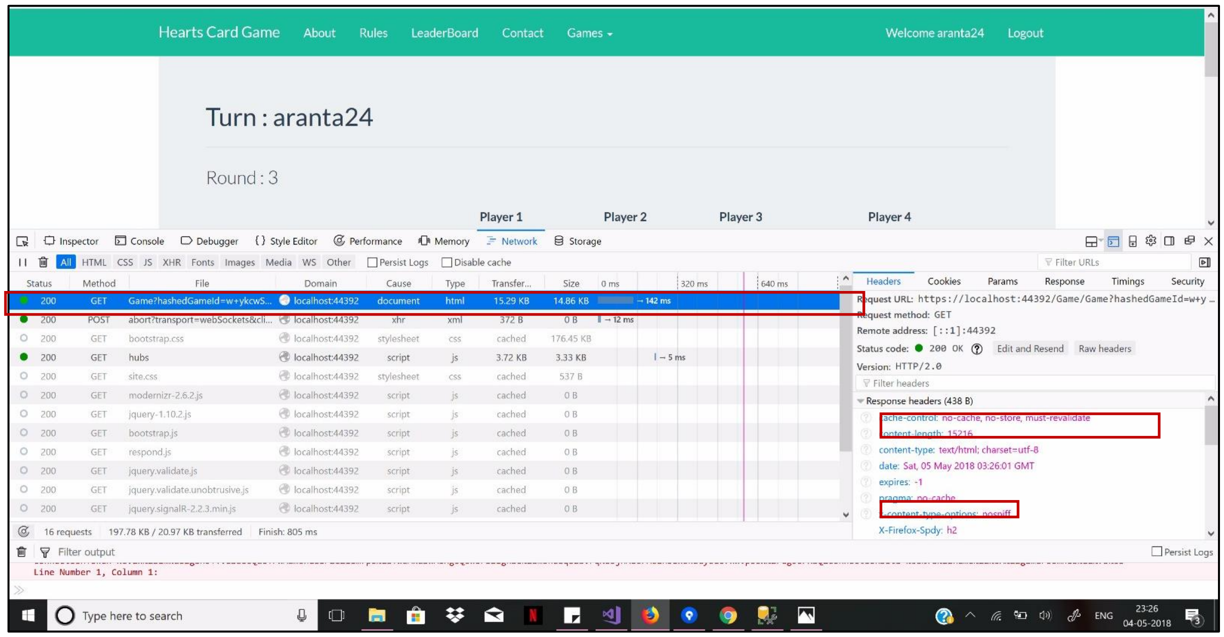

Hearts.DAL (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [8724] iisexpress.exe
Global.asax.cs UserDAL.cs Error.cshtml Web.config Login.cshtml AccountController.cs HomeController.cs
Hearts.MVC
Hearts.MVC.MvcApplication
Application_Start()
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List Autos Locals Watch 1
Ready Ln 23 Col 56 Ch 56 INS
HeartsCardGame master

```
0 references | Aranta Rokade, 3 days ago | 1 author, 5 changes
public class MvcApplication : System.Web.HttpApplication
{
    0 references | Aranta Rokade, 3 days ago | 1 author, 3 changes | 0 exceptions
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
        //Register AuditTrail Attribute
        GlobalFilters.Filters.Add(new AuditTrailFilter());
        //Remove X-AspNetMvc-Version Response Header
        MvcHandler.DisableMvcResponseHeader = true;
    }

    0 references | Aranta Rokade, 7 days ago | 1 author, 3 changes | 0 exceptions
    protected void Application_PreSendRequestHeaders()
    {
        //Remove Server Header
        Response.Headers.Remove("Server");
        //Remove X-AspNet-Version Header
        Response.Headers.Remove("X-AspNet-Version");
    }
}
}
```

Fig. 6.10.2. Http Response Headers in App Start

6.11. Cache Control

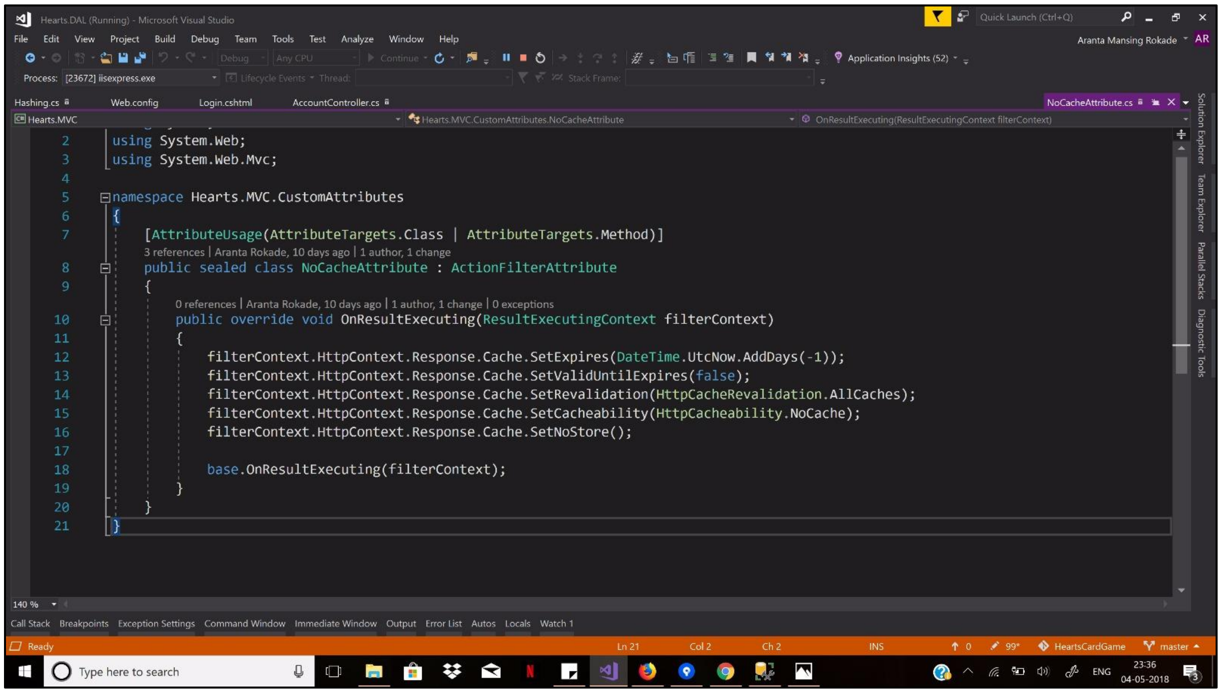
We have enforced no cache control in our application by creating a `[NoCache]` attribute. This attribute can be added on the controllers or actions methods to disable the caching of response. This can be seen in Fig.6.11.1, the GET request for game data is not cached whereas the other calls to .js and .css files are cached.



Status	Method	File	Domain	Cause	Type	Transfer...	Size
200	GET	Game?hashedGameId=w+ykcwS...	localhost:44392	document	html	15.29 KB	14.86 KB
200	POST	abort?transport=webSockets&cli...	localhost:44392	xhr	xml	372 B	0 B
200	GET	bootstrap.css	localhost:44392	stylesheet	css	cached	176.45 KB
200	GET	hubs	localhost:44392	script	js	3.72 KB	3.33 KB
200	GET	site.css	localhost:44392	stylesheet	css	cached	537 B
200	GET	modernizr-2.6.2.js	localhost:44392	script	js	cached	0 B
200	GET	jquery-1.10.2.js	localhost:44392	script	js	cached	0 B
200	GET	bootstrap.js	localhost:44392	script	js	cached	0 B
200	GET	respond.js	localhost:44392	script	js	cached	0 B
200	GET	jquery.validate.js	localhost:44392	script	js	cached	0 B
200	GET	jquery.validate.unobtrusive.js	localhost:44392	script	js	cached	0 B
200	GET	jquery.signalR-2.2.3.min.js	localhost:44392	script	js	cached	0 B

```
16 requests   197.78 KB / 20.97 KB transferred   Finish: 805 ms

Request URL: https://localhost:44392/Game/Game?hashedGameId=w+y ...
Request method: GET
Remote address: [::1]:44392
Status code: 200 OK
Version: HTTP/2.0

Response headers (438 B)
cache-control: no-cache, no-store, must-revalidate
content-length: 15216
content-type: text/html; charset=utf-8
date: Sat, 05 May 2018 03:26:01 GMT
expires: -1
pragma: no-cache
x-content-type-options: nosniff
X-Firefox-Spdy: h2

Line Number 1, Column 1:
```

Fig. 6.11.1. No caching



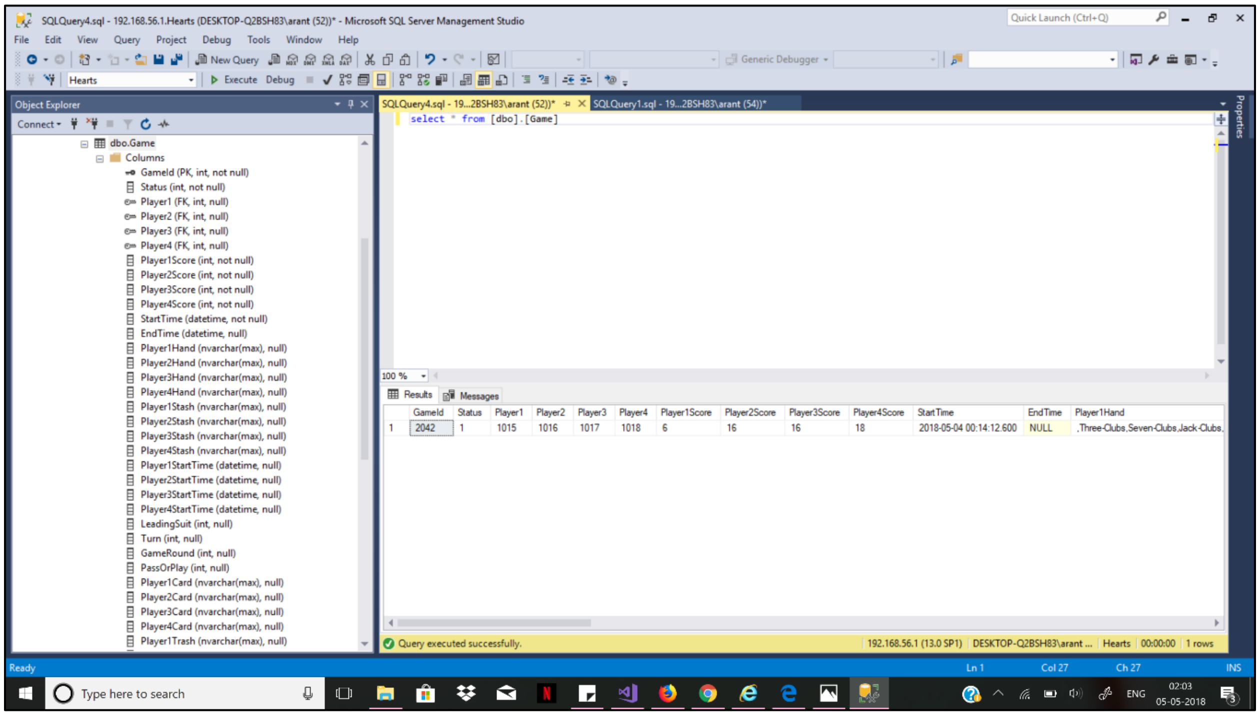
```
using System.Web;
using System.Web.Mvc;

namespace Hearts.MVC.CustomAttributes
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method)]
    3 references | Aranta Rokade, 10 days ago | 1 author, 1 change
    public sealed class NoCacheAttribute : ActionFilterAttribute
    {
        0 references | Aranta Rokade, 10 days ago | 1 author, 1 change | 0 exceptions
        public override void OnResultExecuting(ResultExecutingContext filterContext)
        {
            filterContext.HttpContext.Response.Cache.SetExpires(DateTime.UtcNow.AddDays(-1));
            filterContext.HttpContext.Response.Cache.SetValidUntilExpires(false);
            filterContext.HttpContext.Response.Cache.SetRevalidation(HttpCacheRevalidation.AllCaches);
            filterContext.HttpContext.Response.Cache.SetCacheability(HttpCacheability.NoCache);
            filterContext.HttpContext.Response.Cache.SetNoStore();

            base.OnResultExecuting(filterContext);
        }
    }
}
```

Hearts.DAL (Running) - Microsoft Visual Studio
Aranta Mansing Rokade
Hashing.cs
Web.config
Login.cshtml
AccountController.cs
NoCacheAttribute.cs
Hearts.MVC
Hearts.MVC.CustomAttributes.NoCacheAttribute
OnResultExecuting(ResultExecutingContext filterContext)
Ready
Ln 21
Col 2
Ch 2
INS
HeartsCardGame
master

Fig. 6.11.2 [NoCache] Attribute

6.12. Gameplay Rule Validations

The game restricts user to make certain moves. We have ensured that users are not permitted to make invalid moves. This check is applied on the server so that malicious users cannot cheat. The entire gameplay updates are saved in the database. Which makes it easier to validate the moves on the server.


SQLQuery4.sql - 192.168.56.1.Hearts (DESKTOP-Q2BSH83\arant (52))* - Microsoft SQL Server Management Studio

Object Explorer

dbo.Game
- Columns
  - GameId (PK, int, not null)
  - Status (int, not null)
  - Player1 (FK, int, null)
  - Player2 (FK, int, null)
  - Player3 (FK, int, null)
  - Player4 (FK, int, null)
  - Player1Score (int, not null)
  - Player2Score (int, not null)
  - Player3Score (int, not null)
  - Player4Score (int, not null)
  - StartTime (datetime, not null)
  - EndTime (datetime, null)
  - Player1Hand (nvarchar(max), null)
  - Player2Hand (nvarchar(max), null)
  - Player3Hand (nvarchar(max), null)
  - Player4Hand (nvarchar(max), null)
  - Player1Stash (nvarchar(max), null)
  - Player2Stash (nvarchar(max), null)
  - Player3Stash (nvarchar(max), null)
  - Player4Stash (nvarchar(max), null)
  - Player1StartTime (datetime, null)
  - Player2StartTime (datetime, null)
  - Player3StartTime (datetime, null)
  - Player4StartTime (datetime, null)
  - LeadingSuit (int, null)
  - Turn (int, null)
  - GameRound (int, null)
  - PassOrPlay (int, null)
  - Player1Card (nvarchar(max), null)
  - Player2Card (nvarchar(max), null)
  - Player3Card (nvarchar(max), null)
  - Player4Card (nvarchar(max), null)
  - Player1Trash (nvarchar(max), null)

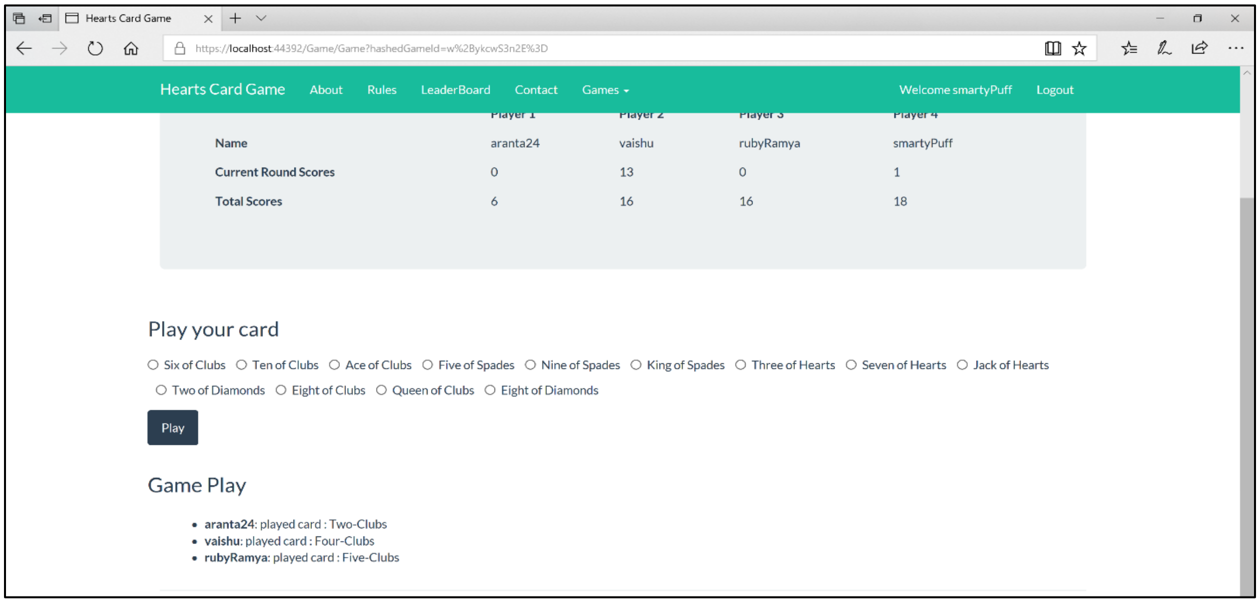
```
select * from [dbo].[Game]
```

	GameId	Status	Player1	Player2	Player3	Player4	Player1Score	Player2Score	Player3Score	Player4Score	StartTime	EndTime	Player1Hand
1	2042	1	1015	1016	1017	1018	6	16	16	18	2018-05-04 00:14:12.600	NULL	,Three-Clubs,Seven-Clubs,Jack-Clubs,

Query executed successfully. 192.168.56.1 (13.0 SP1) DESKTOP-Q2BSH83\arant ... Hearts 00:00:00 1 rows

Fig. 6.12.1. Current game state in database.

Players are notified of the game updates using SignalR push notifications. See the figure below.


https://localhost:44392/Game/Game?hashedGameId=w%2BykcwS3n2E%3D

Hearts Card Game About Rules LeaderBoard Contact Games Welcome smartyPuff Logout

	Player 1	Player 2	Player 3	Player 4
Name	aranta24	vaishu	rubyRamya	smartyPuff
Current Round Scores	0	13	0	1
Total Scores	6	16	16	18

Play your card

○ Six of Clubs ○ Ten of Clubs ○ Ace of Clubs ○ Five of Spades ○ Nine of Spades ○ King of Spades ○ Three of Hearts ○ Seven of Hearts ○ Jack of Hearts ○ Two of Diamonds ○ Eight of Clubs ○ Queen of Clubs ○ Eight of Diamonds

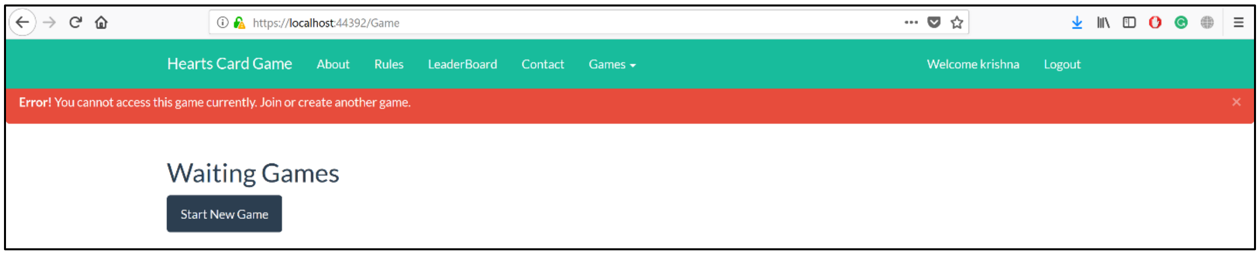
Play

Game Play

- aranta24: played card : Two-Clubs
- vaishu: played card : Four-Clubs
- rubyRamya: played card : Five-Clubs

Fig. 6.12.2. Gameplay updates.

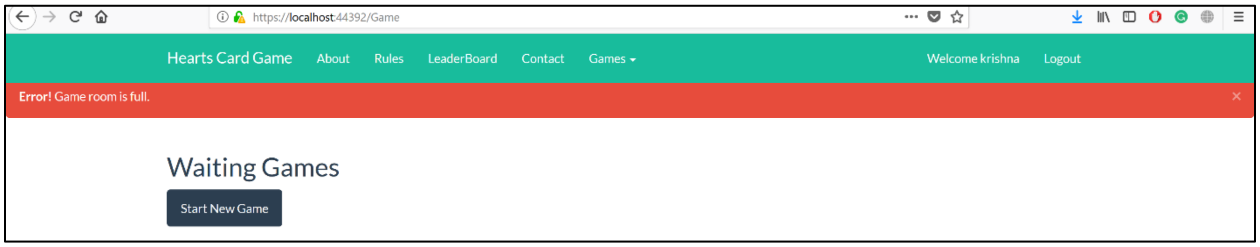
Some of the gameplay validations are show in the figures below in this section.


https://localhost:44392/Game

Error! You cannot access this game currently. Join or create another game.

Waiting Games

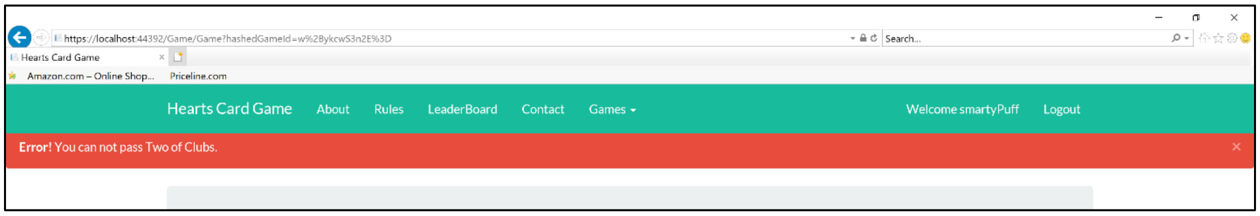
Fig.6.12.1. A player cannot view a game that is in progress.


https://localhost:44392/Game

Error! Game room is full.

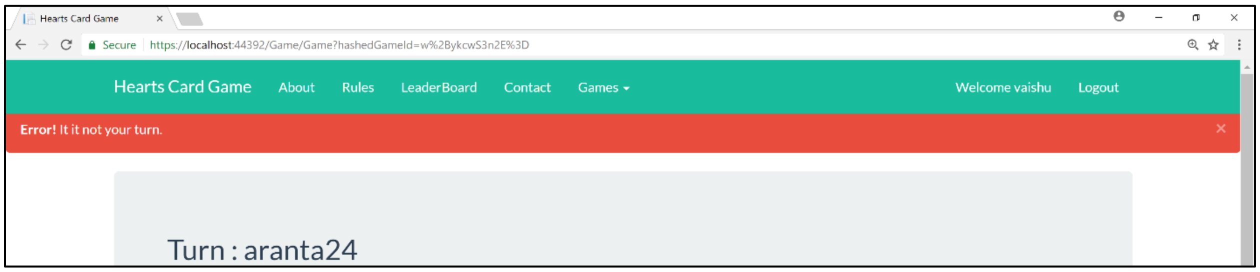
Waiting Games

Fig. 6.12.2. A player cannot join a game that is full.


https://localhost:44392/Game/Game?hashedGameId=w%2BykcwS3n2E%3D

Error! You can not pass Two of Clubs.

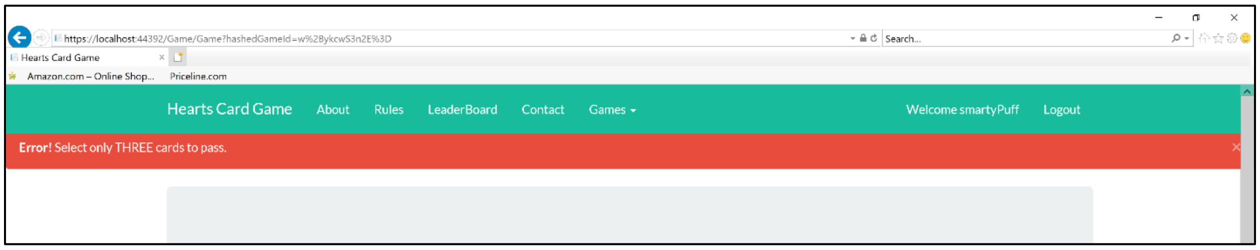
Fig. 6.12.3. A player is not permitted to pass Two of Clubs as trash


https://localhost:44392/Game/Game?hashedGameId=w%2BykcwS3n2E%3D

Error! It it not your turn.

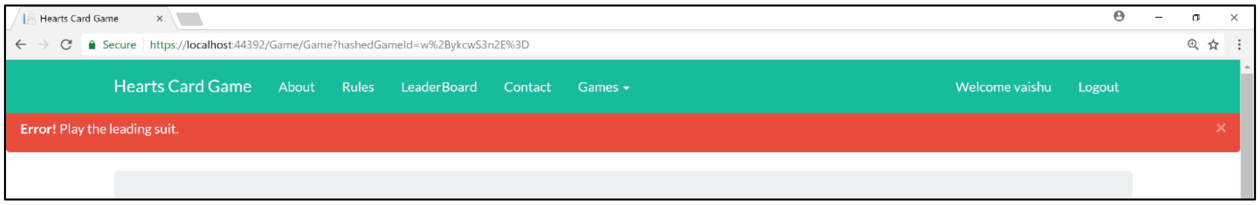
Turn : aranta24

Fig. 6.12.4 A player cannot play out of turn.


https://localhost:44392/Game/Game?hashedGameId=w%2BykcwS3n2E%3D

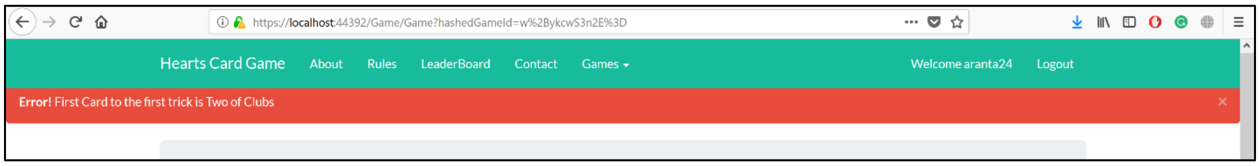
Error! Select only THREE cards to pass.

Fig. 6.12.5. Exactly three cards have to be passed as trash.



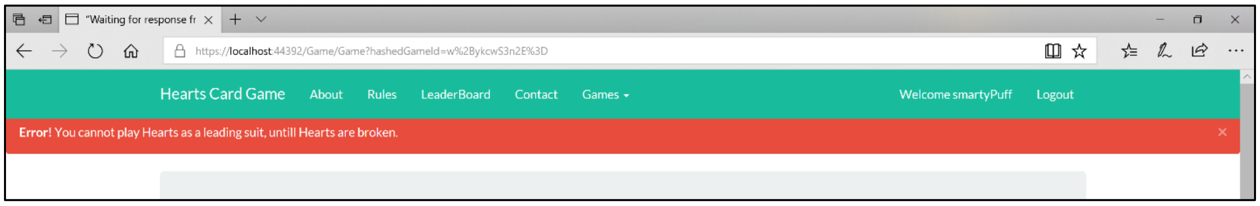
Error! Play the leading suit.

Fig. 6.12.6. A player cannot play another suit if he/she has the leading suit in his/her hand.



Error! First Card to the first trick is Two of Clubs

Fig. 6.12.7. The opening card to the first trick of every round is Two of Clubs.



Error! You cannot play Hearts as a leading suit, untill Hearts are broken.

Fig. 6.12.8. Hearts cannot be lead unless Hearts are broken by a trick.

6.13. OWASP ZAP Testing

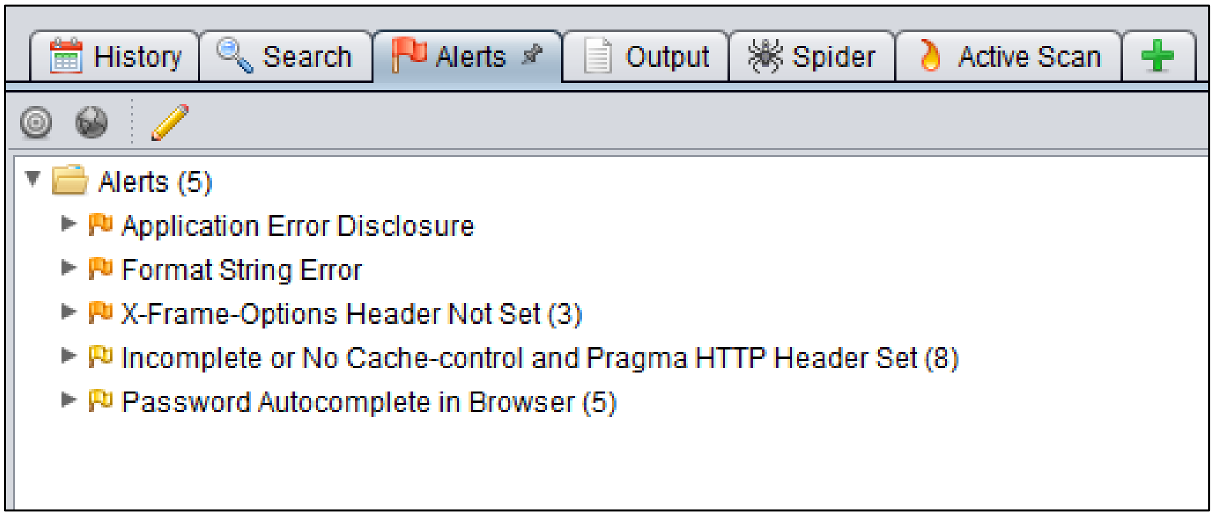
The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing.

ZAP provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.

The following is the screenshot of the testing we have conducted:

6.13.1. Standard Mode – 04/24/2018

Medium Level attacks – 3, Low Level attacks – 2. Refer Fig. 6.12.1 to see the details.

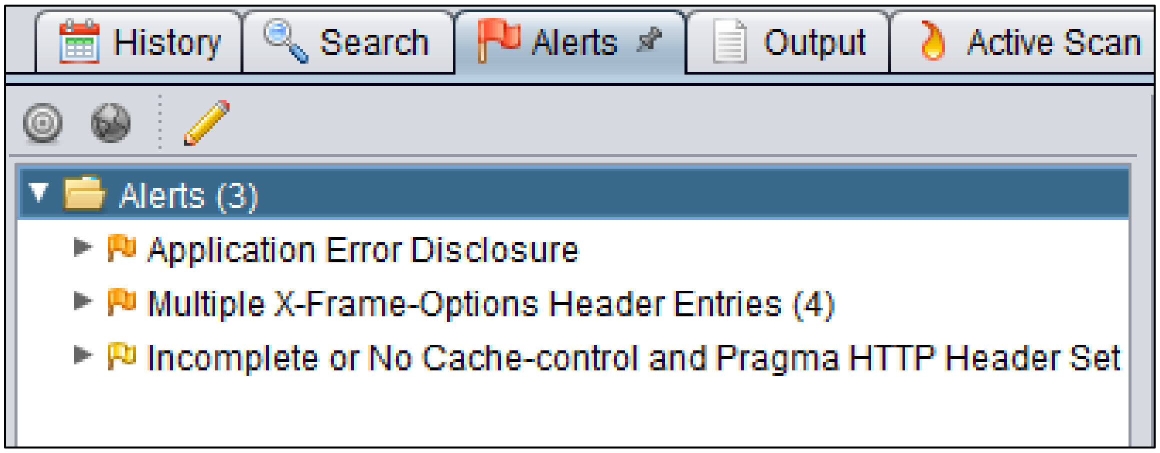


Alerts (5)
- Application Error Disclosure
- Format String Error
- X-Frame-Options Header Not Set (3)
- Incomplete or No Cache-control and Pragma HTTP Header Set (8)
- Password Autocomplete in Browser (5)

Fig. 6.12.1. Standard Mode OWASP ZAP Test

6.13.2. Attack Mode – 05/04/2018

Medium Level attacks – 2, Low Level attacks – 1. Refer Fig. 6.12.2 to see the details.



Alerts (3)
- Application Error Disclosure
- Multiple X-Frame-Options Header Entries (4)
- Incomplete or No Cache-control and Pragma HTTP Header Set

Fig. 6.12.2. Attack Mode OWASP ZAP Test

The measures taken by us to reduce the risks encountered by OWASP-ZAP are:

1. Format String Error (Medium)

Description: This occurred due of lack of character count on the username and email address fields on the registration page.

Mitigation: We removed this risk by adding the minimum and maximum length constraints on the username name and email address, using `[MinLength(6)]` and `[StringLength(20)]` data annotations.

2. Password Autocomplete in browser (Low)

Description: This risk occurred because the autocomplete feature on forms with passwords was not disabled. Thus, allowing the browser to store and retrieve passwords.

Mitigation: We added autocomplete=”off” attribute on the form tags containing passwords fields.

7. Conclusion

This game was developed to learn secure software design and programming. While, developing this game we learnt many aspects of secure programming. Implementation of SSL certificates for secure client server communications, the importance of the HTTP response headers, protecting data in transit and in stationary state, controlling input as well as output data are some of the many highlights associated with this project development.

REFERENCES

[OWASP-DOTNET] Shane, Murnion ;John, Staveley; Steve, Bamelis; Xander, Sherry. ".NET Security Cheat Sheet", OWASP. Last Modified 09-20-2017 https://www.owasp.org/index.php/.NET_Security_Cheat_Sheet

[OWASP-XSS] https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[MSDN-Valinput] "ValidateInputAttribute Class", Microsoft Documentation https://msdn.microsoft.com/enus/library/system.web.mvc.validateinput attribute(v=vs.118).aspx

[Wheeler2017] David, A.Wheeler. "Secure-Software-2-Input-Validation.ppt".10-22-2017

[Wheeler2018] David, A.Wheeler. "Secure-Software-6-Output.ppt".01-24-2018

[Wheeler-Lec8] David, A.Wheeler. "Secure-Software-8-Errors.ppt".10-25-2017

[WIKI-Hearts] https://en.wikipedia.org/wiki/Hearts

[WIKI-TLS] https://en.wikipedia.org/wiki/Transport_Layer_Security

[Andras2015] Andras,Nemes. "How to enable SSL for a .NET project in Visual Studio".2015

https://dotnetcodr.com/2015/09/18/how-to-enable-ssl-for-a-net-project-in-visual-studio/

[Bageri2016] Saineshwar, Bageri. "10 Points to Secure Your ASP.NET MVC Application".2016

https://www.codeproject.com/Articles/1116318/Points-to-Secure-Your-ASP-NET-MVC-Applications

[OWASP-SecMisconf] https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

[OWASP-CSRF] https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

[OWASP-SQL] https://www.owasp.org/index.php/SQL_Injection

[Dierks2008] T. Dierks; E. Rescorla. "The Transport Layer Security (TLS) Protocol, Version 1.2". 2008

https://tools.ietf.org/html/rfc5246