A close-up of the sun

Description automatically generated with low confidenceShape, arrow

Description automatically generatedA picture containing text, clipart

Description automatically generated

This User guide presents the capstone project on SQL Attacks Detection and Prevention with the help of a Web Application showing Login Application built with React and Spring boot Restful API.

SQL Attacks Detection and Prevention

Project: Milestone 4: User Guide

Student’s Name: Mubasher Mehnaz Begum

Grand Canyon University

CST-590-0500-Computer Science Capstone Project

Instructor Name: Aiman Darwiche

Detection of SQL injection Attacks and Prevention

User Guide

02/2023

Version 1.2

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 02/10/2023 | 1.1 | Revised Implementation approach | Mubasher M Begum |
| 02/22/2023 | 1.2 | Revised user guide | Mubasher M Begum |

Copyright Notice

Original work Copyright 2023 Mubasher Mehnaz Begum

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Preface - About This Manual**

# Document Structure

This document consists of a User Guide for the Capstone project on SQL Detection and Prevention application. This manual is only intended for the explanation and use of the SQL Dection and prevention application developed for education and awareness purpose of programmers or IT Professionals working on Web Technologies.

The User Guide is intended for those who are unfamiliar with the SQL attacks and need a place to start. In general this guide presents an overview and the theory of the SQL attacks and prevention mechanisms that can help block such kind of threats to any web applications relying on SQL databases for data. It also contains some of the more advanced subjects which are more topically oriented.

# Intended Audience

This manual is intended for both beginning and experienced users of the web technologies.

Table of Contents

[Introduction 5](#_Toc127135188)

[1.0 Purpose 5](#_Toc127135189)

[2.0 Overview 5](#_Toc127135190)

[3.0 System Summary 5](#_Toc127135191)

[4.0 System Configuration 6](#_Toc127135192)

[5.0 Getting Started to install the tools required 6](#_Toc127135193)

[6.0 Starting the Application 7](#_Toc127135194)

[7.0 Using the System: UI application 7](#_Toc127135195)

[8.0 Testing the API call to Java REST API from React 8](#_Toc127135196)

[9.0 Troubleshooting 8](#_Toc127135197)

[10.0 Help and contact details 8](#_Toc127135198)

# Introduction

SQL Injection attacks (or SQLi) alter SQL queries and carried out by injecting malicious code by exploiting application vulnerabilities.  Bypassing validation on the feeble login page permits us to get to the information base and take client information without knowing their secret word. All data in the data set can be recovered just utilizing union’s based assault strategy.

# Purpose

The goal of this project is to create a safe channel for user transactions. Its goal is to prevent SQL injection when running queries against the database and to safeguard the database. Using a unique static and dynamic analytic approach to detect SQL injection threats. This approach eliminates SQL query metadata at execution (nonlinear system) and contrasts those to SQL queries previously examined (static method).

# Overview

The objective of this task is to fabricate a solid web application that can consequently recognize and hinder SQL infusion assaults. To do this, we can join methods to recognize and moderate SQL infusion assaults.

In this project, several strategies for detecting and preventing SQL injection attacks are tested. Input validation, stored procedures, parameterized statements, and other protection strategies are used to thwart attacks.

# System Summary

The proposed solution for Login bypass page, has entry point as User Interface that is built in React with JavaScript and bootstrap for styling.

**The project has following components:**

* Frontend Application: Which presents the User Interface for the Login application. This is written in React (JavaScript).
* Java Rest API: Utilizes Java Spring boot framework for developing a RESTful API that React application calls for “/authenticate” POST call. This middleware connects with My SQL
* Database: The Database for the application is MSySQL. It contains table for Users to retrieve a list of users and their roles.

# System Configuration

Front end Login Page

Graphical user interface, application

Description automatically generated

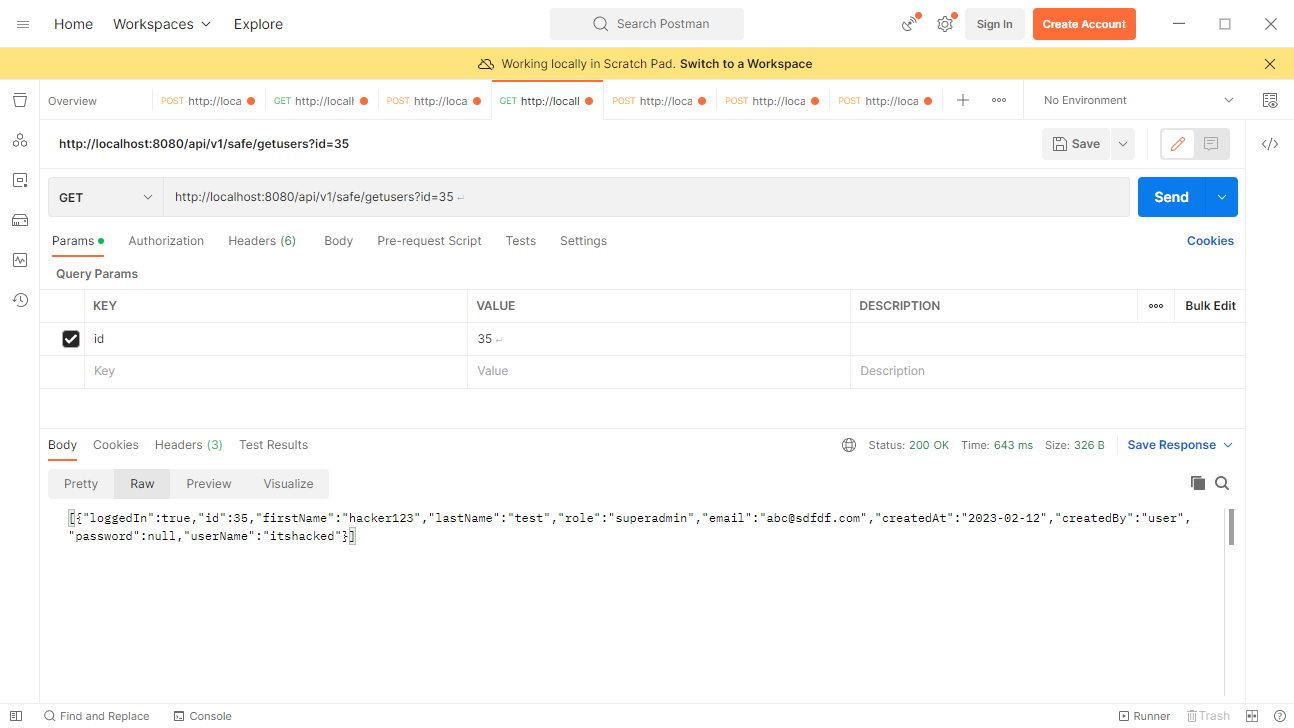
# Getting Started to install the tools required

Install the following Tools & Open source libraries:

* Open-source Java SDK
* IntelliJ Idea Community version
* Open-source Microsoft Visual Studio Code
* Microsoft MySQL Installed, MySQL Workbench
* Postman
* Node JS, Node package Manager
* React, JavaScript, HTML5, CSS3
* Spring boot, Hibernate, MySQL driver, Maven
* MySQL server & MySQL workbench

# Starting the Application

* Start MySQL server through workbench
* Create the schema and database tables using the database script provided
* Insert few record into users table
* Extract the Source code provided
* From IntelliJ Idea, load the java-api folder and perform mvn clean install that installs all the maven dependencies
* Run the Java Springboot API using: **mvn spring-boot:run** in the terminal
* The rest API is running on:
  + <http://localhost:8080>
  + Validate API using postman by performing a Get request using end point: <http://localhost:8080/api/v1/safe/getusers?id=35>



# Using the System: UI application

* Running React Application:
  + Open Visual studio Code and open the UI-code folder.
  + Install dependencie by doing “npm install” in the terminal
  + Start the application by entering: “npm start”
  + The application runs on <http://localhost:300>
  + Testing the application using any browser, navigate to <http://localhost:3000>
  + Enter username and password in the input fields provided.

# Testing the API call to Java REST API from React

Enter data into username/password, the API returns response from database layer.

Graphical user interface

Description automatically generated

# Troubleshooting

* Make sure the MySQL server service is started and local instance is running through workbench.
* Make sure the Java REST API is running on port 8080 to test the API call
* Make sure the react app runs on port 3000

# Help and contact details

Contact Mubasher M Begum through email: [mbegum@my.gcu.edu](mailto:mbegum@my.gcu.edu)