South-Eastern Finland
University of Applied Sciences

Algirdas Kvedaravičius

Smart authentication and authorization system

Bachelor's thesis
Degree

Degree Programme in Information Technology

2021

South-Eastern Finland
University of Applied Sciences

Author (authors)	**Degree title**	**Time**
Algirdas Kvedaravičius	Bachelor of Engineering	April 2021
Thesis title		
Smart authentication and authorization system		38 pages 1 page of appendices
Commissioned by		
Supervisor		
Timo Hynninen		

Abstract

The main goal of this thesis is to create a smart security system that will reduce the risk of hacking and theft using Internet of Things (IoT) technology. During the project, a prototype system was designed that will use several types of authentications and will authorize employees to enter the workplace while registering it in the database.

Cybernetic attacks and theft of inventory and information are always a significant harm to any company, because it results in additional expenses that sometimes can lead into a bankruptcy if damage is severe enough. With smart security technology, we can ensure a lower risk factor for the stolen important information, materials or equipment.

This project produced a prototype of a possible fully working system. The prototype is very versatile, but it is still a bit slow and inefficient but could easily be improved with proper funding. Additional security features could also be installed for an even lower risk factor.

Keywords

authentication, authorization, security system, internet of things, thesis, Arduino

Contents

Table of figures

1 INTRODUCTION

The main goal of this thesis is to create a smart security system that will reduce the risk of hacking and theft using Internet of Things (IoT) technology. During this project, a prototype system will be designed that will use several types of authentications and will authorize employees to enter the workplace while registering it in the database.

1.1 Relevance of the topic

Cybernetic attacks and theft of inventory and information are always a significant harm to any company, because it results in additional expenses that sometimes can lead into a bankruptcy if damage is severe enough. (Girdhar, Anup, and Navneet Kaur Popli 2017)

With smart security using IoT technology, we can ensure a lower risk factor for the stolen important information, materials or equipment.

1.2 Objective

The objective is to create a prototype system that recognizes employees who want to enter the workplace, verifies whether or not they have access and registers them in a database, that can be checked at any time through a web server.

1.3 Tasks

In order to achieve the objective, the following tasks should be carried out:

- Configure and create MySQL database.
- Plan the operating scheme of the entire system.
- Start a web server where all system information can be accessed.

- Implement the system.
- Test the prototype.

The thesis will be structured into 3 main parts: Chapter 2 will analyze components, programming languages and security measurements of the system. Chapter 3 is the practical part where the prototype is built, and Chapter 4 where it is tested.

2 Research part

Cybercrime and thefts affect all sizes of organizations from the biggest one to the smallest. Important Information theft is the main factor and it includes information like: customers data, organization products, research analysis, financial data etc. These attacks will not ever stop because there is profit in information, according to Cybersecurity Ventures, damages from cybercrime will reach 6 trillion USD.

In order to protect themselves all organizations have to invest into security measures to fight this threat. Option varies from very expensive advanced system to more affordable simpler versions. This way expenses from damage will be lower.

To prevent damage an affordable smart security system must be developed that records the exact date on which authorized personnel, identified by fingerprint and RFID scanners, enter the workplace and when the workplace is locked. The locked room will be additionally protected by motion sensors that will notify the system of unauthorized movement in the locked workplace. Web server must have security measures against cyberattacks.

The system will use five programming languages: PHP, HTML, SQL, Arduino, Java. The “Xampp” application will be used to create the local server, in which will be hosted a Web server and a MySQL database. The data transfer uses a “Processing“ application that will help transfer sensor data from Arduino to a web server via a universal serial bus (USB) connection, thus the system will operate regardless of the Internet connection. The Arduino microcontroller is used to read sensor data. The website will use the HTTPs protocol to protect against external threats for server security.

2.2 Microcontroller

A microcontroller is a small computer that is designed to perform one programmed function. All microcontrollers have a central processing unit (CPU) that contains integrated random-access memory (RAM).

These microcontrollers are great devices for simple tasks such as: reading sensors data, transferring information or performing simple functions. Since these devices are affordable than stronger computers, they are heavily used in the field of the Internet of Things.

Table 1 Microcontroller comparison.

	Microcontrollers	
	8051 microcontroller	Arduino
Input voltage	5 V to 6.6 V	6 V to 20 V
GPIO contacts output voltage	5 V and Max 15 mA	5 V or 3.3 V max 50mA
Memory	Flash memory: 8 kB RAM: 256 Bits	Flash memory: 32 kB RAM: 2 kB
Clock speed	12 MHz	16 MHz can be boosted up to 20 MHz
Programming software	Uvision IDE	Arduino IDE
GPIO contacts	32 contacts	20 contacts

Arduino

Arduino is an easy-to-use microcontroller with free software This microcontroller is extremely flexible: it gives the desired voltage, has integrated resistors and LEDs, more affordable and faster compared to other microcontrollers. Arduino IDE software is supported on many operating systems: Windows, Macintosh OSX, Linux. Other microcontroller software usually works only with the Windows operating system.

8051 microcontroller

The 8051 microcontroller is an 8-bit microcontroller that features energy efficiency and integrated touch screen equipment. As a result of these features, the 8051 microcontroller is heavily used in the media, gaming, mobile phones and automotive sector.

Uvision IDE is a rather complex software that can perform many different functions but is therefore not friendly to new users. Also, the Uvision IDE is only supported on the windows operating system.

This project will use an Arduino microcontroller due to its flexibility, ease of use and more affordable price.

2.3 Xampp

Xampp is a free software designed to easily create a web server. This program has Apache, MySQL, FileZilla, Mercury, Tomcat, and many other useful services. Xampp software frequently updates Apache, MySQL PHP and Perl services, which makes it possible to work reliably and without interference.

The Apache module service will be used to create a web server because it supports the PHP programming language.

PhpMyAdmin software that manages the MySQL Database will also be used.

2.4 Security System Information Subsystem

The smart security system fully functions on the internal network and without an external Internet connection, due to increased protection. An external Internet connection is required to connect from the outside.

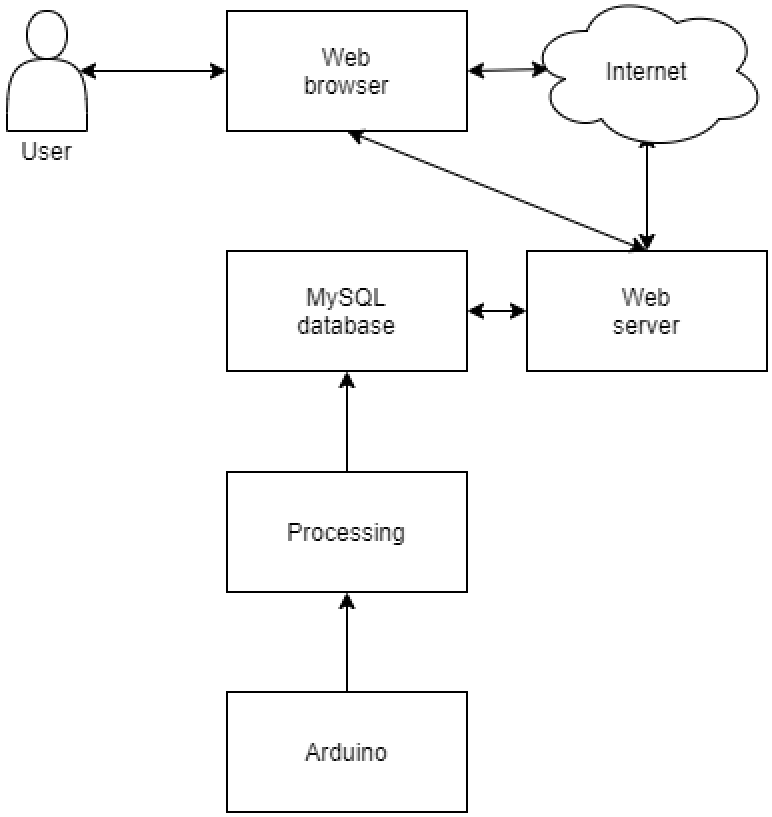


Figure 1 A diagram of a smart security logical connection.

As we can see, these programs will be used:

- Arduino IDE – the only Arduino software.
- Processing – best software to transfer information from Arduino via a USB connection.
- PhpMyAdmin- will be used to control database as it is integrated in the Xampp software.
- Apache – will be used as a web server for the same reason (integrated Xampp software), and is also very stable and popular, which makes it an excellent choice.
- A browser is everyone's choice because logging in is available from all browsers.

Remote desktop connection application is needed to connect to a web server remotely, that will be used to connect to any work computer on the local network.

2.5 Analysis of software languages

The system uses a variety of software that will also use different programming languages. These software use their own unique programming languages that are similar or written for another popular programming language, as appropriate.

Arduino software can be programmed in any programming language, but the Arduino programming language is based on C and C++ languages. This programming language looks and writes on a C/C++ basis and uses many of the features of these languages. Also uses C/C++ compilers.

Processing software uses java programming language, with simplifications that add more mathematical functions and operators. Also processing software has a graphical user interface that facilitates compilation and execution stages. Java is a common purpose programming language that is object-oriented. This language is one of the most popular applications for Android applications and client-server websites.

Back-end web server will use PHP programming language. PHP – one of the most popular general purpose programming language that is most suitable for website development work. Since this language is best for back-end web development, it will be used to create an Apache web server. The part of the code that will be written in this language will be responsible for communicating between the Web server and the database. HTML programming language will be used in the front-end development process. This language will be responsible for everything the user sees and in creating tables.

2.6 Security analysis

In the security system, the primary priority is protection. The first line of defense in this security system is access to the system only through a local network, but it is not protection without flaws. To better protect the system from external threats, it is necessary to first analyze possible malicious attacks and how it can be protected from threats.

2.6.1 Cross Site Scripting

XSS is a protection vulnerability that allows you to insert malicious code into a web page that can be viewed by other users. This usually happens when text boxes on web pages allow you to save text using HTML codes and are stored by a web server. In this way, the hackers can put the script in the text box and when another user opens a web page with a saved malicious code, the hacker receives all the information about the user. In 2017, nearly 40% of attacks on website servers were carried out on such a principle. (M. Mohammadi, B. Chu and H. R. Lipford 2017).

The easiest way to prevent this vulnerability is to delete text fields or embed text filters that prevent HTML codes from being saved.

2.6.2 SQL injection attack

SQL injection attack(SQLIA) – most common threat to a database in which the hacker inserts malicious string of code into input box, to gain access or make changes to data inside database. This dangerous because information inside a database could be deleted or stolen.(Alwan, Zainab S., and Manal F. Younis 2017)

To prevent SQL attacks, you must connect to the MySQL database without administrator rights and install filters on the sign-in form.

2.6.3 DDoS attack

DDoS attacks are a cyberattack that use bots and send multiple requests until the system is finally broken, because if there is a lack of RAM and CPU, the power system crashes and other users can no longer connect. There are several ways to protect yourself from DDoS attacks:

- Buy stronger servers.
- Install anti-DDoS software.
- Configure routers against DDoS attacks.

Since this security system is located in the internal network, it is practically impossible to touch it from the outside using a DDoS attack.

2.7 Research conclusion

The security system will be stored in the internal network due to additional protection against various attacks, as well as independence from the external Internet network connection. Also, the web server will additionally be protected by the HTTPs protocol. Xampp software will be used to create the Web server: PhpMyAdmin MySQL database, Apache web Server. Codes for software will be written in: PHP, HTML, Java, and Arduino languages. Arduino microcontroller will be used for reading sensor data. Processing application will be used to read data from Arduino, process it and send it to the web server to be stored in the MySQL database.

3 Project

In order for the system to work, you need to write the necessary software codes so that the information is retrieved from the sensors to the users. You also need to prepare all the necessary applications and configure the MySQL database.

3.1 Prototype model

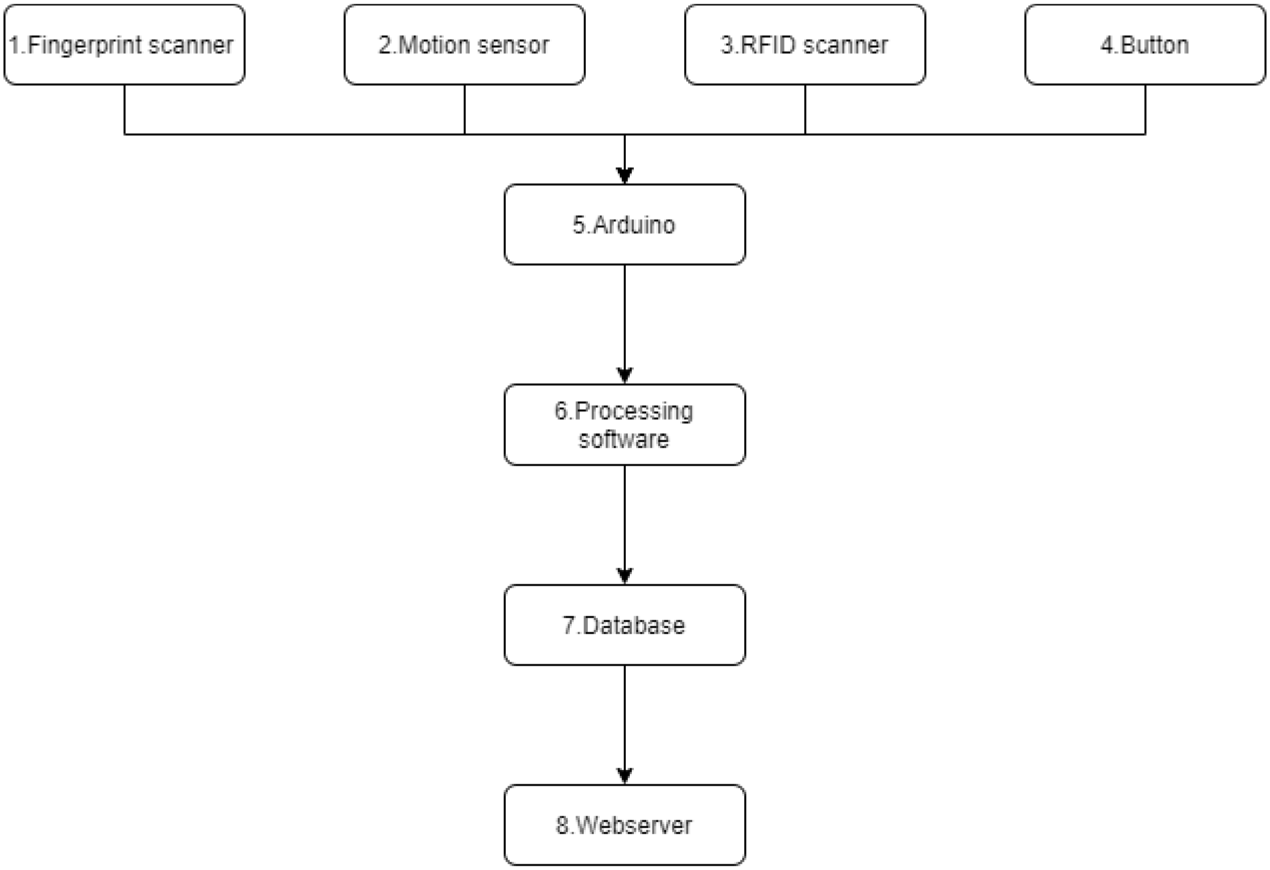


Figure 2 System principle scheme

The diagram (see figure 2) contains the numbered main components of the system and their functions:

1. Fingerprint scanner – this sensor is responsible for identifying and authorizing employees.
2. Motion sensor – protection against physical intrusions, detects movement in a locked room.
3. RFID scanner – used as workplace “key”, correct magnetic key is required to unlock a workplace.
4. Button - for locking the workplace.

5. Arduino – a microcontroller that recycles all the information received from sensors and unlocks the door of the workplace.
6. Processing software – designed to retrieve information from Arduino and place it in a database
7. Database – contains all information from Arduino and user logins.
8. Web server – workplace security system information is accessed through this server.

The table lists which Arduino connections will be used in the implementation of the project.

Table 2 Arduino connections

Component Name	Arduino outputs	Component inputs
Processing software	USB	USB
Fingerprint sensor	GND	GND
Fingerprint sensor	5V	5V
Fingerprint sensor	Digital 3	RX
Fingerprint sensor	Digital 2	TX
Motion sensor	GND	GND
Motion sensor	5V	5V
Motion sensor	Digital 4	OUT
RFID sensor	GND	GND
RFID sensor	3.3V	3.3V
RFID sensor	Digital 9	RST
RFID sensor	Digital 10	SDA
RFID sensor	Digital 11	MOSI
RFID sensor	Digital 12	MISO
RFID sensor	Digital 13	SCK
Button	GND	GND
Button	Digital 7	OUT

3.2 Network topology

Designed workplace topology model (see Figure 3) that would allow information to get from sensors to users. Arduino and sensors are connected by IoT corresponding wires. Arduino and workstation containing Processing software are connected by a USB cable. Rest of the network is connected by an ethernet cable.

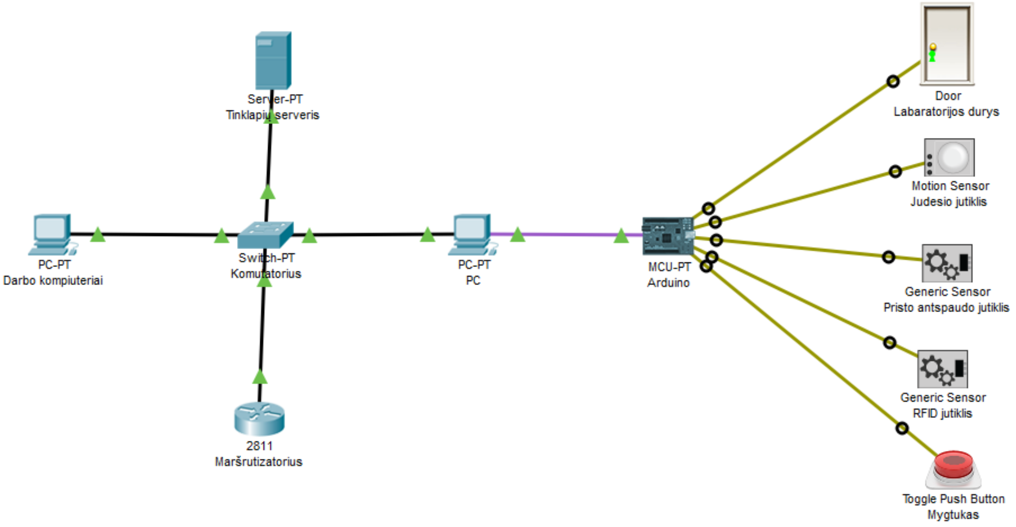


Figure 3 Network Topology

3.3 Database

Database must be configured to accommodate a full data from sensors. Registered employees must also be listed, as well as login codes.

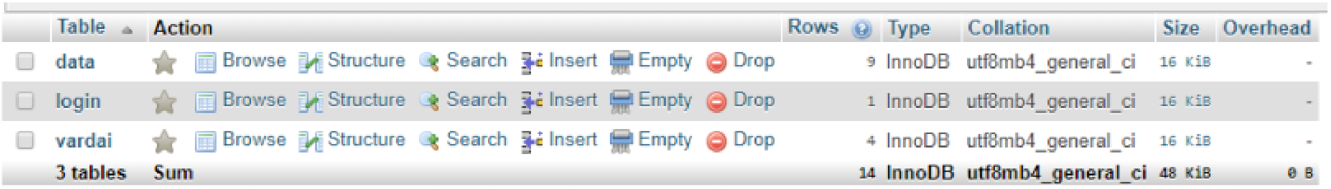
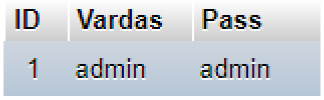


	Table	Action						Rows	Type	Collation	Size	Overhead
□	data	Browse	Structure	Search	Insert	Empty	Drop	9	InnoDB	utf8mb4_general_ci	16 KiB	-
□	login	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16 KiB	-
□	vardai	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_general_ci	16 KiB	-
	3 tables	Sum						14	InnoDB	utf8mb4_general_ci	48 KiB	0 B

Figure 4 Database structure

Three tables are created that contain hosted information:

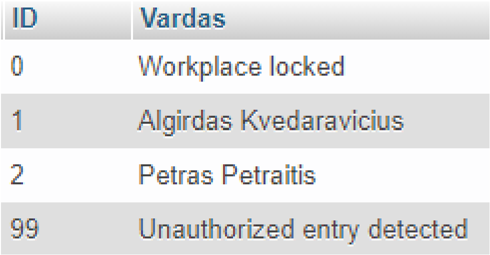
1. “Data”.
2. “Login”.
3. “Vardai”.



ID	Vardas	Pass
1	admin	admin

Figure 5 “Login” table structure

“Login” table stores administrator logins.



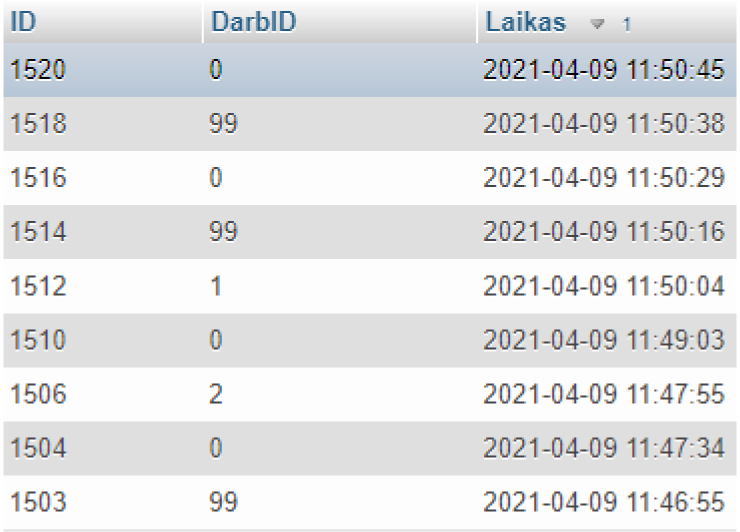
ID	Vardas
0	Workplace locked
1	Algirdas Kvedaravicius
2	Petras Petraitis
99	Unauthorized entry detected

Figure 6 “vardai” table structure

The table “vardai” stores information about what each code means:

1. 0 – workplace locked

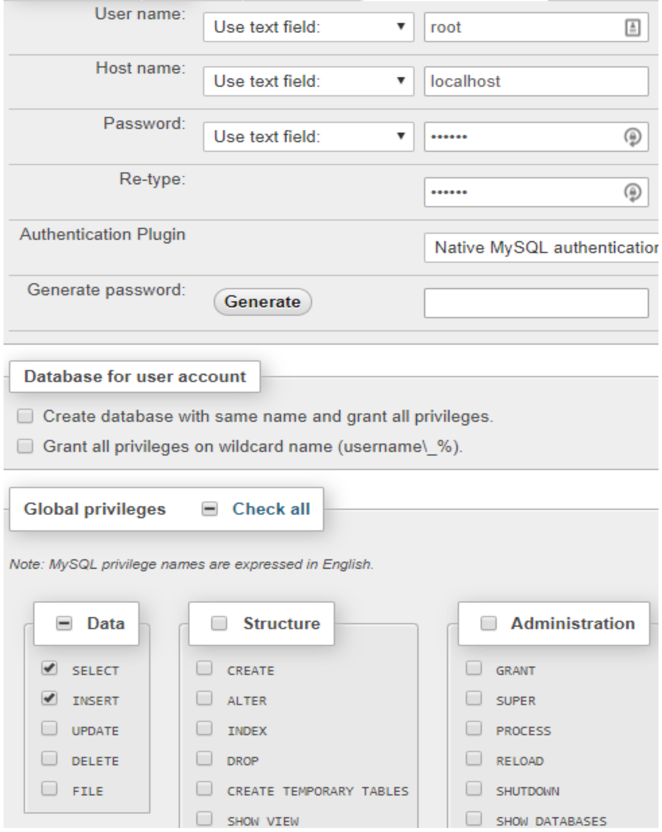
- 99 – break-in detected
- 1 to 98 – Registered users.



ID	DarbID	Laikas ▼ 1
1520	0	2021-04-09 11:50:45
1518	99	2021-04-09 11:50:38
1516	0	2021-04-09 11:50:29
1514	99	2021-04-09 11:50:16
1512	1	2021-04-09 11:50:04
1510	0	2021-04-09 11:49:03
1506	2	2021-04-09 11:47:55
1504	0	2021-04-09 11:47:34
1503	99	2021-04-09 11:46:55

Figure 7 “data” table structure

The table “Data” stores the received codes and when they were saved to the database with an accuracy of seconds.



User name: Use text field: root

Host name: Use text field: localhost

Password: Use text field: ••••••

Re-type: ••••••

Authentication Plugin: Native MySQL authenticatio

Generate password: Generate

Database for user account

□ Create database with same name and grant all privileges.

□ Grant all privileges on wildcard name (username_%).

Global privileges ⊟ Check all

Note: MySQL privilege names are expressed in English.

⊟ Data
- ☑ SELECT
- ☑ INSERT
- □ UPDATE
- □ DELETE
- □ FILE

□ Structure
- □ CREATE
- □ ALTER
- □ INDEX
- □ DROP
- □ CREATE TEMPORARY TABLES
- □ SHOW VIEW

□ Administration
- □ GRANT
- □ SUPER
- □ PROCESS
- □ RELOAD
- □ SHUTDOWN
- □ SHOW DATABASES

Figure 8 MySQL database user creation

Creates a database user who can only embed or retrieve information to and from the database, thereby increasing protection against an SQL injection attack.

3.4 Data model

In this chapter data models will be displayed.

3.4.1 Data Flow Chart

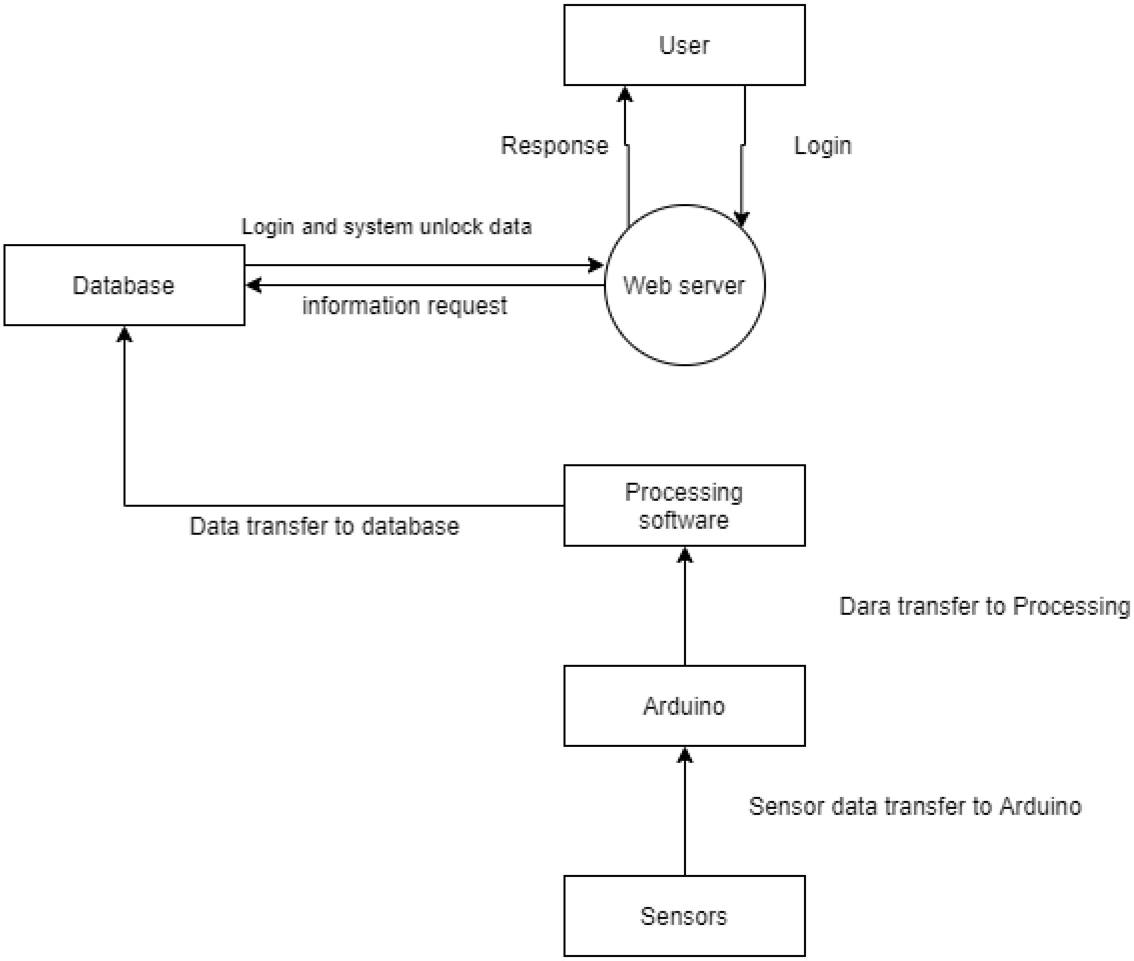

User
Response
Login
Login and system unlock data
Database
Web server
information request
Processing software
Data transfer to database
Dara transfer to Processing
Arduino
Sensor data transfer to Arduino
Sensors

Figure 9 DFD Level 0 Chart

In the chart that has been drawn (see figure 9), we see how the data flows in the system.

3.4.2 User Activity Diagram

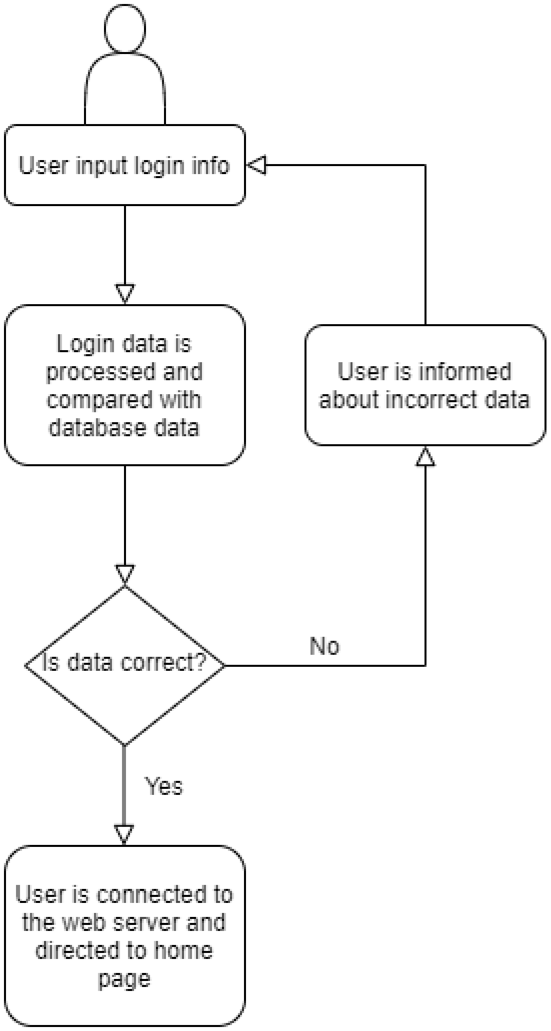

User input login info
Login data is processed and compared with database data
User is informed about incorrect data
Is data correct?
No
Yes
User is connected to the web server and directed to home page

Figure 10 Diagram of user sign-in activity.

The diagram (see figure 10) shows employee's system login activity.

3.5 Graphical user interface model

Here is login (see figure 11 and table 3) and home (see figure 12 and table 4) pages graphical user interfaces.

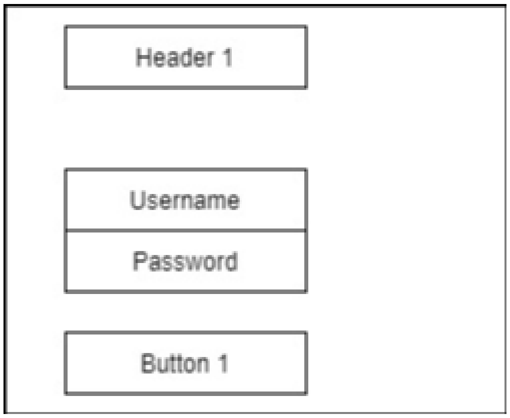


Figure 11 Login Page Graphical User Interface

Table 3 login page graphical user interface specification

Field	**Field type**
Header 1	Name
Username	Text
Password	Password
Button 1	submit

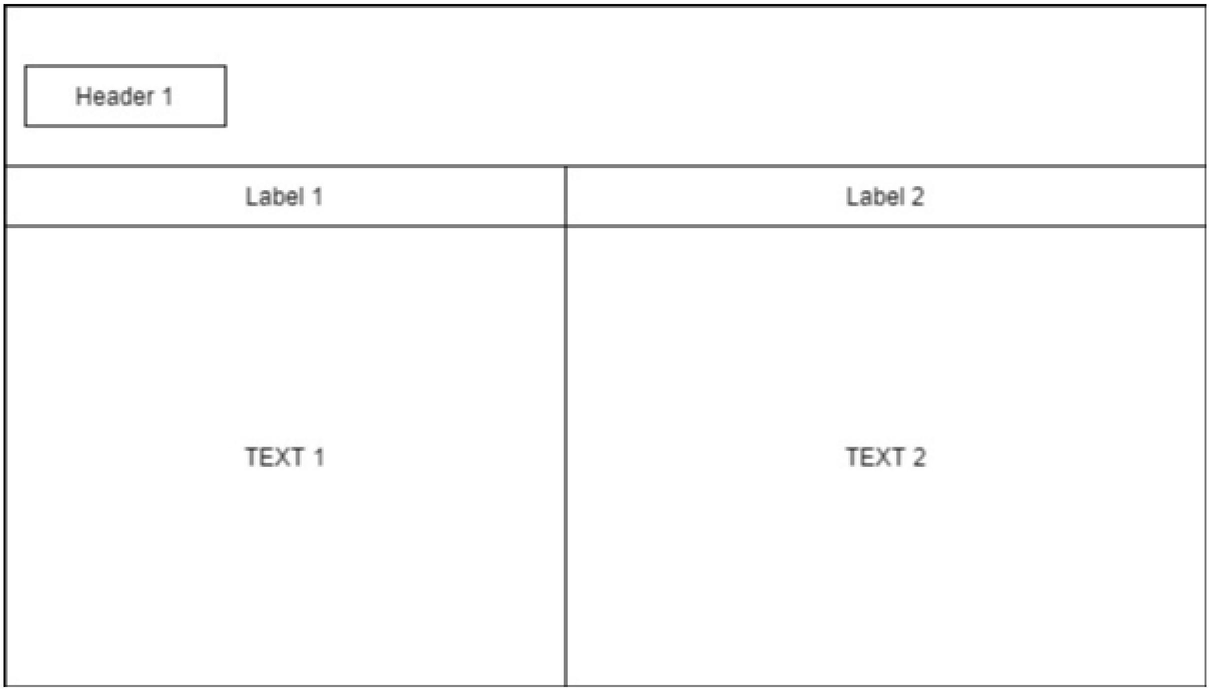


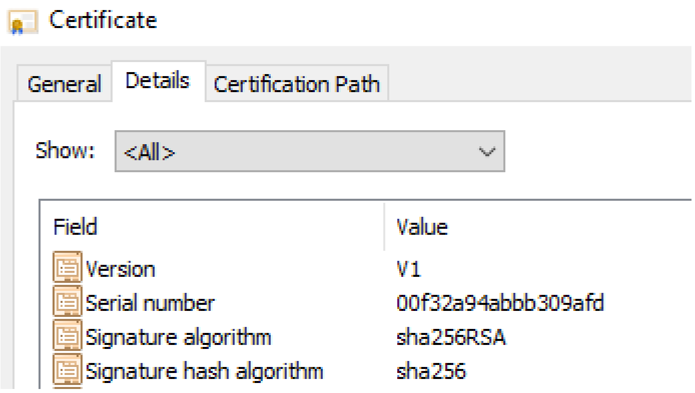
Figure 12 Home Page Graphical User Interface

Table 4 Home Graphical User Interface specification

Field	**Field type**
Header 1	Name
Label 1	Column Name
Label 2	Column Name
Text 1	Text
Text 2	Text

3.6 Apache Web Server

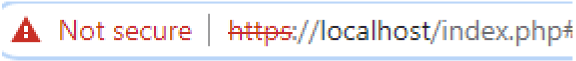
The Xampp software creates an Apache web server that will display when allowed users have entered the laboratory, and when the laboratory is locked. The Web server itself is protected by the HTTPs protocol.



Show: <All>

Field	Value
Version	V1
Serial number	00f32a94abbb309afd
Signature algorithm	sha256RSA
Signature hash algorithm	sha256

Figure 13 HTTPs certificate



Not secure | ~~https~~://localhost/index.php#

Figure 14 Browser search box

The browser does not recognize the manually created certificate, which indicates a "Not secure" message. However, this certificate is a fully functional HTTPs protocol.

3.5.1 "Login" page and software code

To increase security, a Login page is created first to enter the administrator login information, that is placed in the database "login" table.

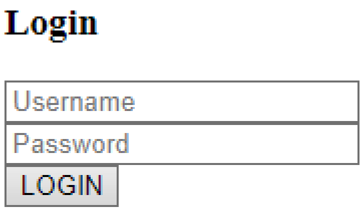
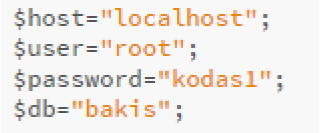


Figure 15 Sign-in form

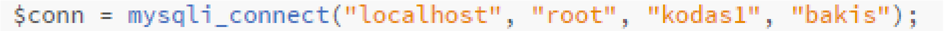
A login.php file is created that is loaded into the C:\xampp\htdocs directory.



```
$host="localhost";
$user="root";
$password="kodas1";
$db="bakis";
```

Figure 16 MySQL data

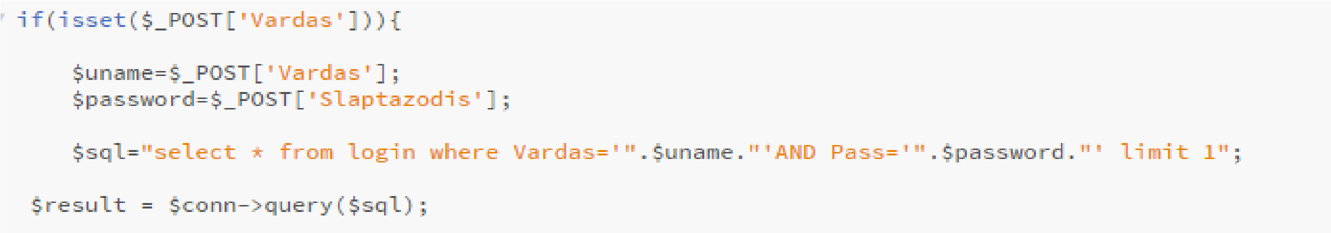
Database login information.



```
$conn = mysqli_connect("localhost", "root", "kodas1", "bakis");
```

Figure 17 MySQL database connection code

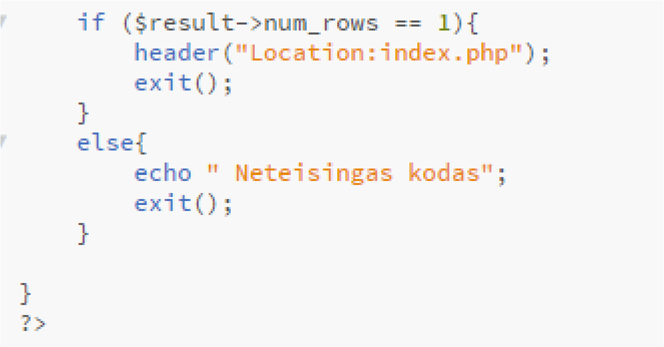
Connecting to database string.



```
if(isset($_POST['Vardas'])){

    $uname=$_POST['Vardas'];
    $password=$_POST['Slaptazodis'];

    $sql="select * from login where Vardas='".$uname."'AND Pass='".$password."' limit 1";

 $result = $conn->query($sql);
```

Figure 18 Login verification code

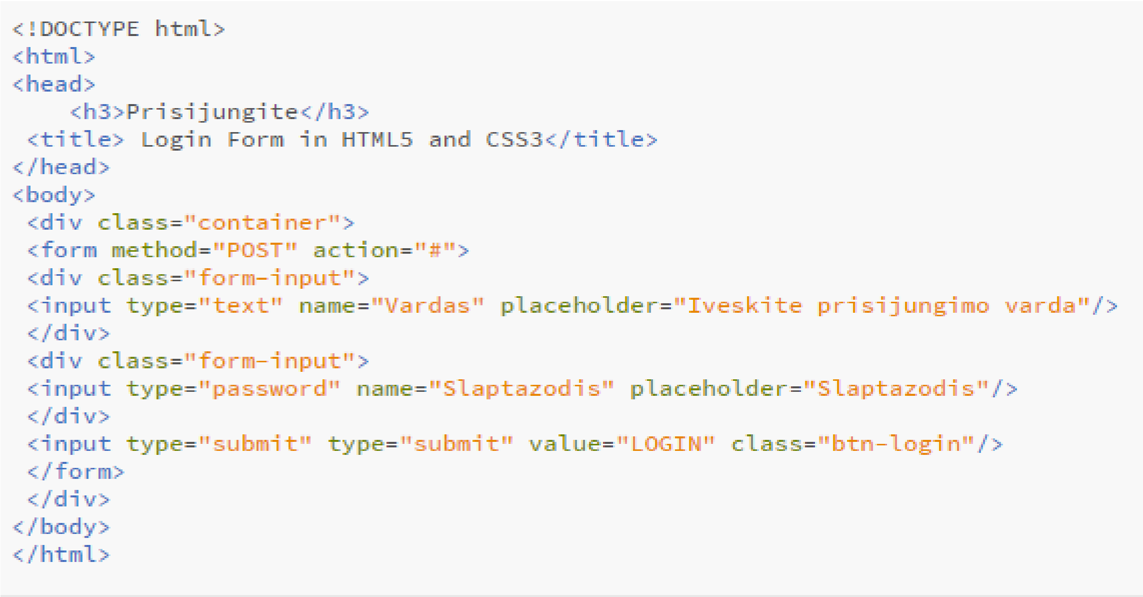
User logins and passwords are checked with the data from the “login” table in the database.



```
    if ($result->num_rows == 1){
        header("Location:index.php");
        exit();
    }
    else{
        echo " Neteisingas kodas";
        exit();
    }

}
?>
```

Figure 19 Page routing code

The login information you enter is checked against the information in the database, if it matches the user is redirected to an “index.php” file. The PHP script is closed in this place.



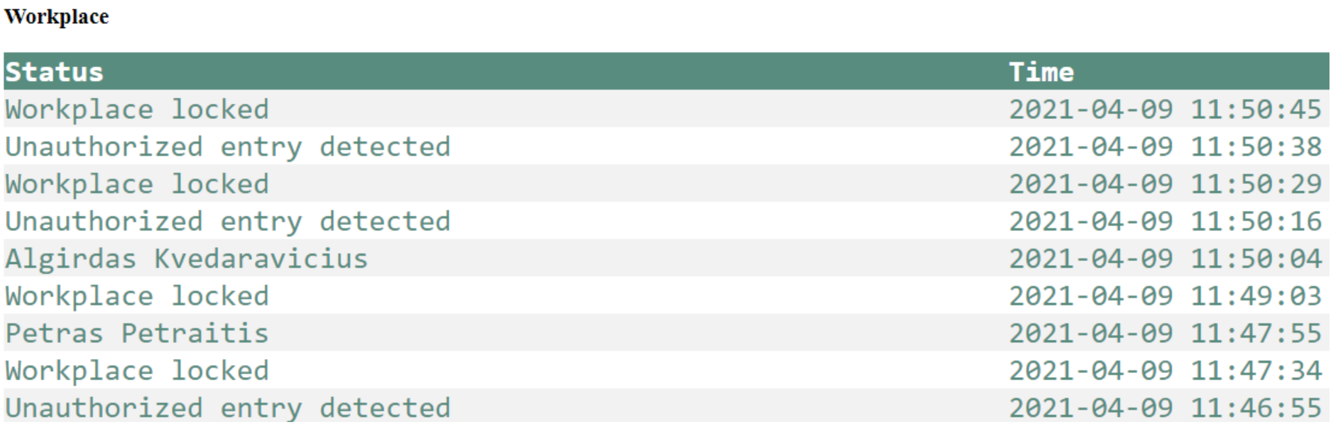
```
<!DOCTYPE html>
<html>
<head>
    <h3>Prisijungite</h3>
 <title> Login Form in HTML5 and CSS3</title>
</head>
<body>
 <div class="container">
 <form method="POST" action="#">
 <div class="form-input">
 <input type="text" name="Vardas" placeholder="Iveskite prisijungimo varda"/>
 </div>
 <div class="form-input">
 <input type="password" name="Slaptazodis" placeholder="Slaptazodis"/>
 </div>
 <input type="submit" type="submit" value="LOGIN" class="btn-login"/>
 </form>
 </div>
</body>
</html>
```

Figure 20 Login form code

In the html code part, a login form is created that scans the entered information and points to the php script.

3.5.2 “Workplace History” page and software code

The workplace history page displays a table that lists what happened in the workplace at a certain time.

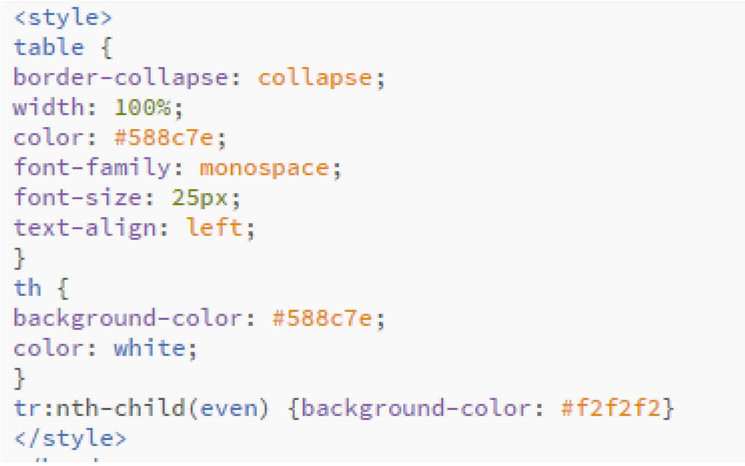


Workplace

Status	Time
Workplace locked	2021-04-09 11:50:45
Unauthorized entry detected	2021-04-09 11:50:38
Workplace locked	2021-04-09 11:50:29
Unauthorized entry detected	2021-04-09 11:50:16
Algirdas Kvedaravicius	2021-04-09 11:50:04
Workplace locked	2021-04-09 11:49:03
Petras Petraitis	2021-04-09 11:47:55
Workplace locked	2021-04-09 11:47:34
Unauthorized entry detected	2021-04-09 11:46:55

Figure 21 Home Page Table

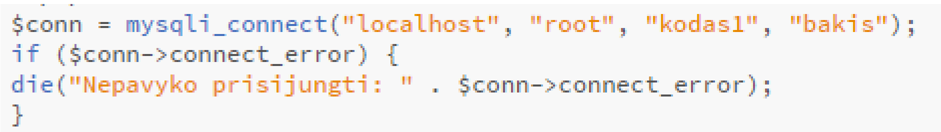
An “index.php” file is created that is loaded into the C:\xampp\htdocs directory.



```
<style>
table {
border-collapse: collapse;
width: 100%;
color: #588c7e;
font-family: monospace;
font-size: 25px;
text-align: left;
}
th {
background-color: #588c7e;
color: white;
}
tr:nth-child(even) {background-color: #f2f2f2}
</style>
```

Figure 22 Table Style Code

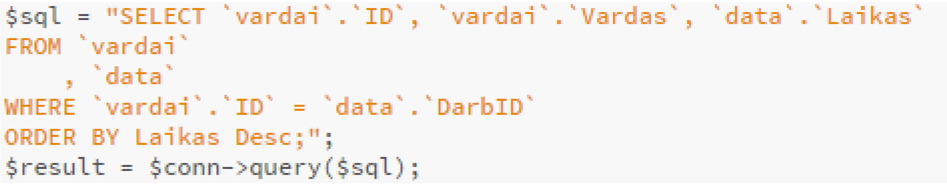
Describes the table and its style.



```
$conn = mysqli_connect("localhost", "root", "kodas1", "bakis");
if ($conn->connect_error) {
die("Nepavyko prisijungti: " . $conn->connect_error);
}
```

Figure 23 MySQL login code

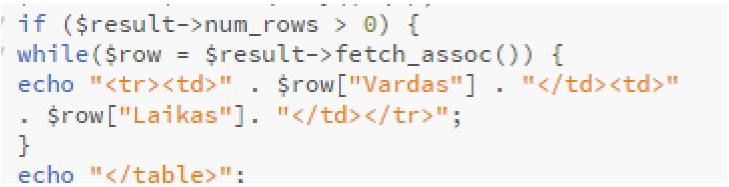
When logging on to the database, if they do not log in, the user will be notified with a message that the connection to the database was unsuccessful.



```
$sql = "SELECT `vardai`.`ID`, `vardai`.`Vardas`, `data`.`Laikas`
FROM `vardai`
    , `data`
WHERE `vardai`.`ID` = `data`.`DarbID`
ORDER BY Laikas Desc;";
$result = $conn->query($sql);
```

Figure 24 SQL Filter Code

The information is taken from two tables: “vardai” and “data”. Codes derived from Arduino are placed in the “data” table, and the meaning of those codes are in the “names” table. The codes are compared, and the table displays employee names instead of numbers. Most recent entries appear at the top.



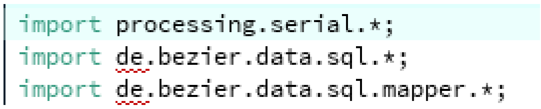
```
if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
echo "<tr><td>" . $row["Vardas"] . "</td><td>"
. $row["Laikas"]. "</td></tr>";
}
echo "</table>";
```

Figure 25 Table input code

A loop is created that fills all columns in a table.

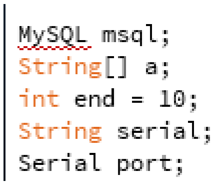
3.7 “Processing” application software code

The Processing application is used to retrieve Arduino information and placing it in the database.



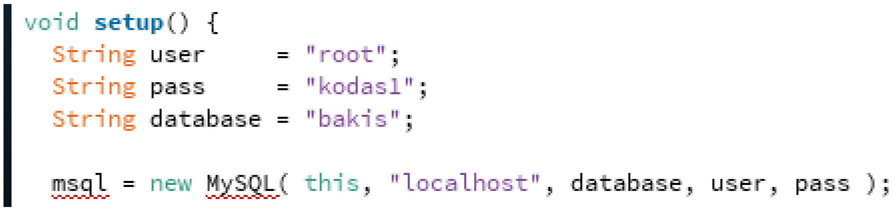
```
import processing.serial.*;
import de.bezier.data.sql.*;
import de.bezier.data.sql.mapper.*;
```

Figure 26 Importing required libraries.



```
MySQL msql;
String[] a;
int end = 10;
String serial;
Serial port;
```

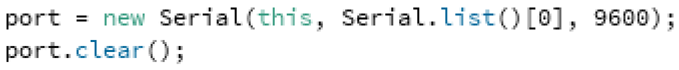
Figure 27 Defined variables



```
void setup() {
  String user     = "root";
  String pass     = "kodas1";
  String database = "bakis";

  msql = new MySQL( this, "localhost", database, user, pass );
```

Figure 28 MySQL login code

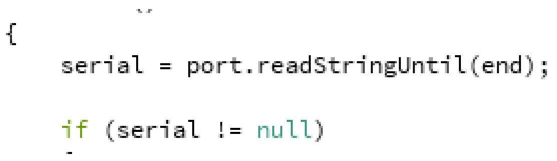
Connects to the database.



```
port = new Serial(this, Serial.list()[0], 9600);
port.clear();
```

Figure 29 Arduino login code

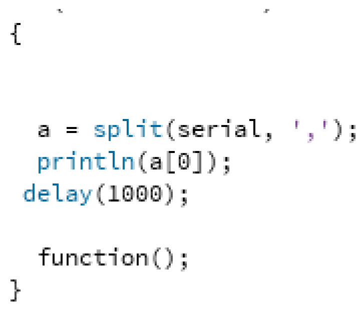
Connects to Arduino via a USB connection and sets compatible speed



```
{
    serial = port.readStringUntil(end);

    if (serial != null)
```

Figure 30 Arduino port read code

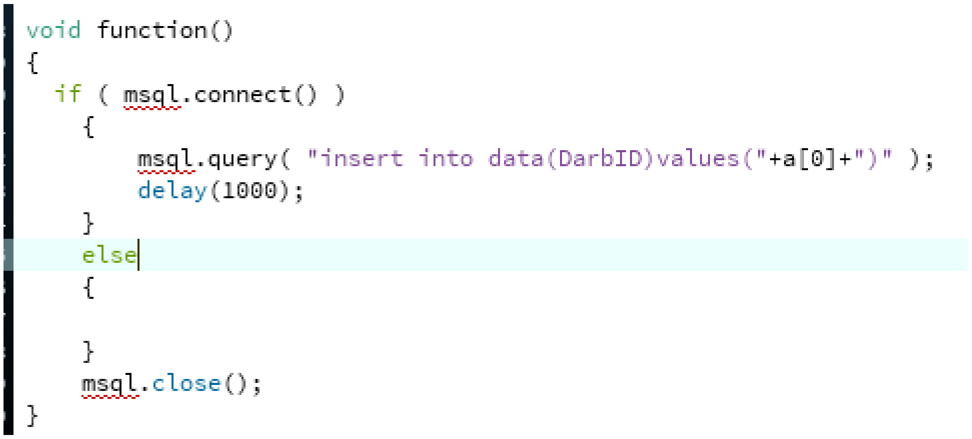
A code which reads the information while there is communication with the Arduino.



```
 {


   a = split(serial, ',');
   println(a[0]);
  delay(1000);

   function();
 }
```

Figure 31 Array code

An array in which the information will be collected, and the resulting information is written to the console.



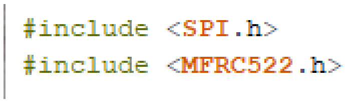
```
void function()
{
  if ( msql.connect() )
    {
        msql.query( "insert into data(DarbID)values("+a[0]+")" );
        delay(1000);
    }
    else
    {

    }
    msql.close();
}
```

Figure 32 Fill in the information database code

The information obtained is placed in the „DarbID” column in the „data” table in the database.

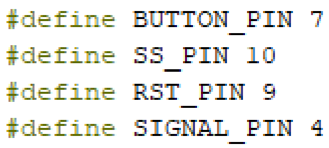
3.8 Arduino IDE Software Code

Arduino software is used to scan sensor information, convert it to code and forward new code Processing application.



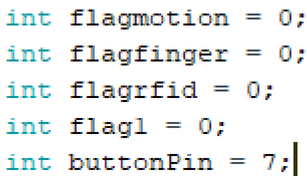
```
#include <SPI.h>
#include <MFRC522.h>
```

Figure 33 Importing required libraries.



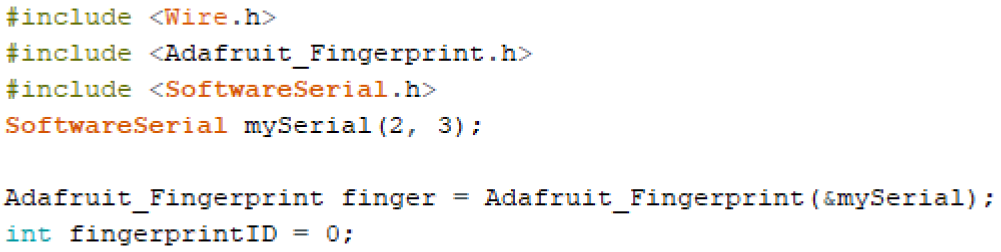
```
#define BUTTON_PIN 7
#define SS_PIN 10
#define RST_PIN 9
#define SIGNAL_PIN 4
```

Figure 34 Defines which connections the sensors are connected to.



```
int flagmotion = 0;
int flagfinger = 0;
int flagrfid = 0;
int flag1 = 0;
int buttonPin = 7;
```

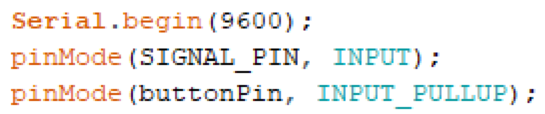
Figure 35 Defines the variables.



```
#include <Wire.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int fingerprintID = 0;
```

Figure 36 Description of the fingerprint sensor.

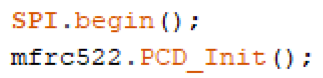
Fingerprint sensor libraries are loaded, sensor is defined and variables are defined.



```
Serial.begin(9600);
pinMode(SIGNAL_PIN, INPUT);
pinMode(buttonPin, INPUT_PULLUP);
```

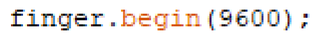
Figure 37 Defines Arduino connections.

The data transfer speed of the Arduino microcontroller shall be determined and the connection mode shall be set. INPUT_PULLUP also activates the built-in Arduino resistor.



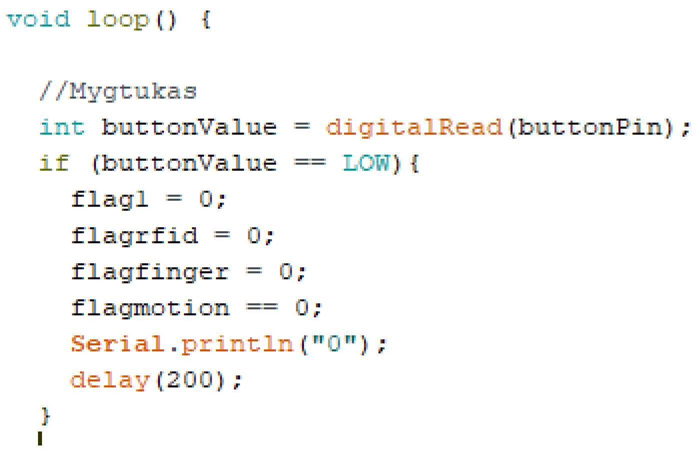
```
SPI.begin();
mfrc522.PCD_Init();
```

Figure 38 Activating RFID sensor.



```
finger.begin(9600);
```

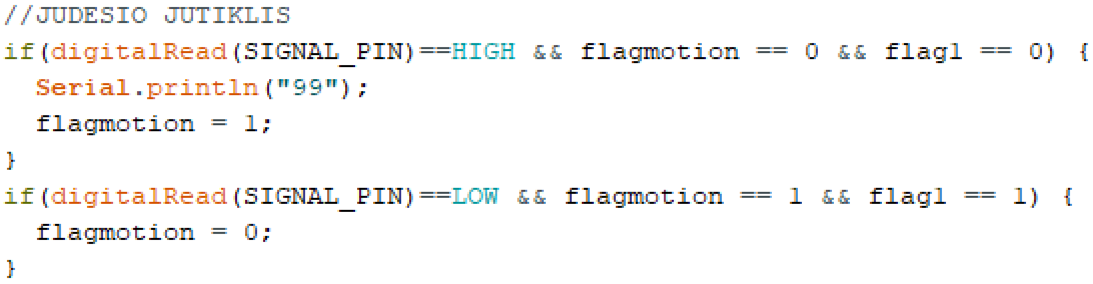
Figure 39 Set compatible fingerprint scanner data transfer speed along with Arduino.



```
void loop() {

  //Mygtukas
  int buttonValue = digitalRead(buttonPin);
  if (buttonValue == LOW){
    flagl = 0;
    flagrfid = 0;
    flagfinger = 0;
    flagmotion == 0;
    Serial.println("0");
    delay(200);
  }
```

Figure 40 Button function code

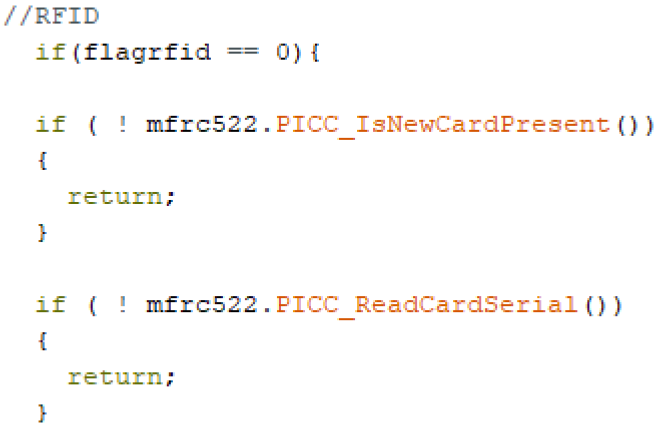
The button function is set in the cycle - when you press a button, it changes the variable stage to 0 and forwards the code to the Processing application: 0. Code 0 means that the workplace is locked.



```
//JUDESIO JUTIKLIS
if(digitalRead(SIGNAL_PIN)==HIGH && flagmotion == 0 && flagl == 0) {
  Serial.println("99");
  flagmotion = 1;
}
if(digitalRead(SIGNAL_PIN)==LOW && flagmotion == 1 && flagl == 1) {
  flagmotion = 0;
}
```

Figure 41 Motion sensor function code

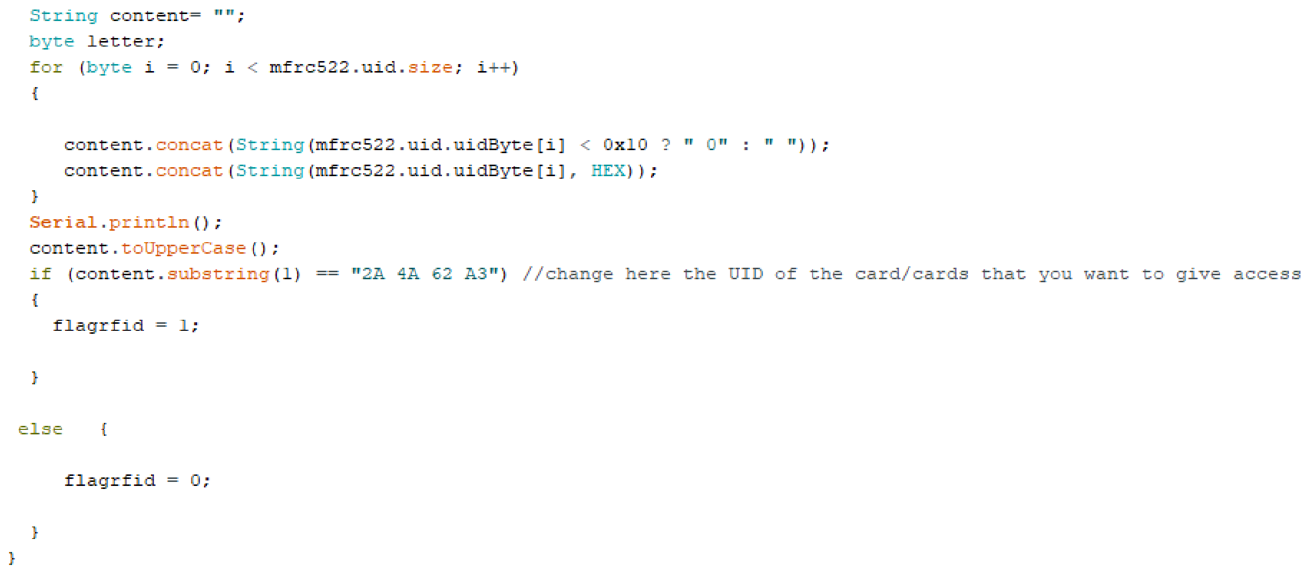
Describes the function of the motion sensor – if the laboratory is locked and the motion sensor detects movement, the Arduino sends the code: 99. Code 99 means that a break-in has been detected in the workplace. “flagmotion” variable is used to send information to database only one time. After unlocking and locking the laboratory, the system can send the information again.



```
//RFID
  if(flagrfid == 0){

  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }

  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
```

Figure 42 RFID magnet key validation code

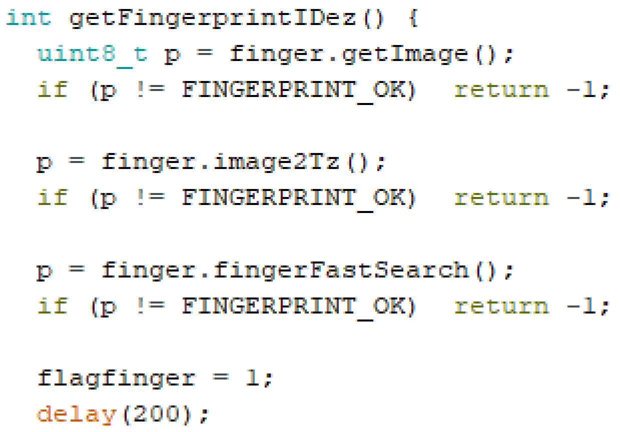
Checks whether the RFID card is on the allowed list.



```
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {

     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  content.toUpperCase();
  if (content.substring(1) == "2A 4A 62 A3") //change here the UID of the card/cards that you want to give access
  {
    flagrfid = 1;

  }

 else   {

     flagrfid = 0;

  }
}
```

Figure 43 RFID sensor function code.

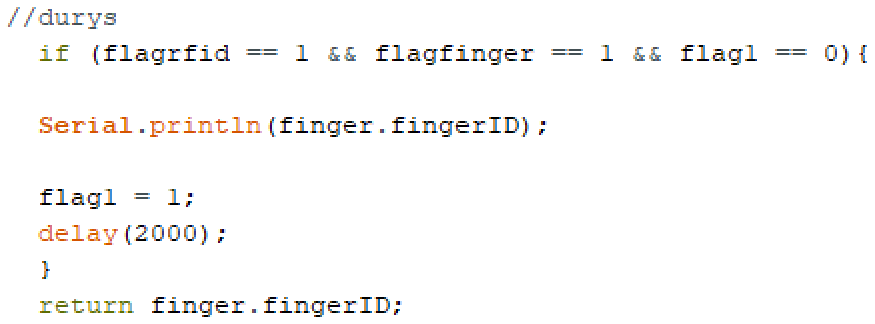
If the RFID magnetic key HEX code is accepted and the „flagrfid" variable state is set to 1, if HEX code is not accepted, the magnetic keys are read further.



```
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  flagfinger = 1;
  delay(200);
```

Figure 44 Fingerprint scanner function code.

Scanned fingerprint is checked for corresponding fingerprints recorded in fingerprint scanner sensor integrated database (up to 128 possible fingerprints), if the fingerprint corresponds to the “flagfinger”, the variable is changed to 1.

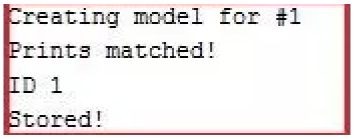


```
//durys
  if (flagrfid == 1 && flagfinger == 1 && flagl == 0){

  Serial.println(finger.fingerID);

  flagl = 1;
  delay(2000);
  }
  return finger.fingerID;
```

Figure 45 Door unlock code

If all the required variables meet the requirements, the laborer door is unlocked and Arduino sends the code to the processing program. The code will be from 1-98 depending on the finger which unlocked the database.

Finally, new employees are added to the fingerprint scanner integrated database. First, enrollment code is launched from File > Examples > Adafruit Fingerprint Sensor Library > Enroll. The desired employee ID and name is entered into the terminal and then “send” button is pressed. “Prints matched” message is received (see figure 46), the employee's fingerprint is included in the fingerprint scanner integrated database. Next, same employee information inputted into MySQL database.



```
Creating model for #1
Prints matched!
ID 1
Stored!
```

Figure 46 Fingerprint registration confirmation

With everything set up, the employee should enter the workplace by:

1. Adding a RFID key to RFID scanner.
2. Adding a finger to the fingerprint scanner.

After finishing the work, the employee presses the button, the door is unlocked for a short period of time and then locks itself.

4 Prototype testing

A test model is created where various functions can be tested.

The first case to be tested is the login page.

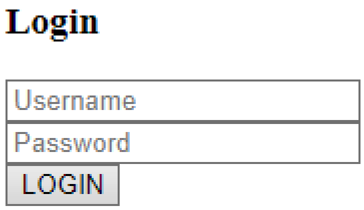


Figure 47 Login page

Table 5 Login page test case

Action	Condition	Result
Pressing the "login" button	Empty fields	Error, message received: "Incorrect username or password"
Pressing the "login" button	Entering incorrect user password	Error, message received: "Incorrect username or password"
Pressing the "login" button	Entering incorrect user Name	Error, message received: "Incorrect username or password"
Pressing the "login" button	Entering only the user's password	Error, message received: "Incorrect username or password"
Pressing the "login" button	Entering only the user's Name	Error, message received: "Incorrect username or password"
Pressing the "login" button	The data entered is correct	Login is successful and user is directed to Home page

The next test case is locking workplace (pressing a button).

Table 6 Button test case

Action	Condition	Result
Pressing a button	Button pressed one time when the workplace is unlocked	The database has been updated and the entry "Workplace locked " is displayed on the home page one time.
Pressing a button	The button is pressed two times when the workplace is unlocked	The database has been updated and the entry "Workplace locked " is displayed on the home page one time.
Pressing a button	The button pressed and held down for a few seconds when the workplace is unlocked	The database has been updated and the entry "Workplace locked " is displayed on the home page one time.
Pressing a button	The button is pressed when the workplace is locked	Nothing happens.
Pressing a button	The button is pressed when the workplace is unlocked	The database has been updated and the entry "Workplace locked " is displayed on the home page one time.

The motion sensor test.

Table 7 Motion sensor testing case

Action	Condition	Result
Hand is moved in the sensor detection range.	The workplace is locked	The database has been updated and the entry " Unauthorized entry detected" is displayed on the Home page
Hand is moved in the sensor detection range.	The workplace is locked, RFID scanned	The database has been updated and the entry " Unauthorized entry detected" is displayed on the Home page
Hand is moved in the sensor detection range.	The workplace is locked, Fingerprint scanned	The database has been updated and the entry "Unauthorized entry

		detected" is displayed on the Home page
Hand is moved in the sensor detection range.	The workplace is unlocked	Nothing happens

The fingerprint scanner test.

Table 8 fingerprint scanner testing case

Action	Condition	Result
Finger scanned	An unregistered finger scanned when the workplace is locked.	Nothing happens
Finger scanned	An unregistered finger scanned when the workplace is unlocked.	Nothing happens
Finger scanned	A registered finger 1 scanned when the workplace is locked and RFID scanned.	The database has been updated and the entry "Algirdas Kvedaravičius" is displayed on the Home page.
Finger scanned	A registered finger 2 scanned when the workplace is locked and RFID scanned.	The database has been updated and the entry "Petras Petraitis" is displayed on the Home page.
Finger scanned	A registered finger 1 scanned when the workplace is unlocked and RFID scanned.	The database has been updated and the entry "Algirdas Kvedaravičius" is displayed on the Home page.
Finger scanned	A registered finger 2 scanned when the workplace is unlocked and RFID scanned.	The database has been updated and the entry "Petras Petraitis" is displayed on the Home page.

Testing is completed and we can see that the prototype functions properly.

5 Conclusion

The main goal of this thesis was to create a smart security system that will reduce the risk of hacking and theft using Internet of Things technology. With thesis projects additional securities implemented in any workplace, the risk of break-ins and theft should be signific ally lowered. It could also be used as monitoring tool to check employees working hours and double check it if important assets are missing during their shift in case of theft.

This project was prototype of a possible fully working system. The prototype is very versatile, but it is still a bit slow, inefficient and it currently can't support large scale workplaces, but it could be easily improved with proper funding. Additional security features could also be installed for even lower risk factor.

References

What is Arduino? (2021) [Accessed 2021-03-20]. Internet access: https://www.arduino.cc/en/Guide/Introduction

Guide to Fingerprint Sensor Module with Arduino (FPM10A) [Accessed 2021-04-03]. Internet access: https://randomnerdtutorials.com/fingerprint-sensor-module-with-arduino/

Security Access using MFRC522 RFID Reader with Arduino [Accessed 2021-04-15]. Internet access: https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/

Arduino with PIR Motion Sensor [Accessed 2021-04-24]. Internet access: https://randomnerdtutorials.com/arduino-with-pir-motion-sensor/

M. Mohammadi, B. Chu and H. R. Lipford, "Detecting Cross-Site Scripting Vulnerabilities through Automated Unit Testing," 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), 2017, pp. 364-373, doi: 10.1109/QRS.2017.46.

Finkenzeller, Klaus. (2010). RFID handbook (electronic resource): Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication (3rd ed.). Oxford: Wiley-Blackwell.

Alwan, Zainab S., and Manal F. Younis. "Detection and prevention of sql injection attack: A survey." International Journal of Computer Science and Mobile Computing 6.8 (2017): 5-17.

Girdhar, Anup, and Navneet Kaur Popli. "Harmful effects of cyber crime in business and economic sustainability." (2017).

Rodriguez, M. (2018). HTTPS Everywhere: Industry Trends and the Need for Encryption. Serials Review, 44(2), 131-137.

Sausalito, Calif. – Nov. 13, 2020. Cybercrime To Cost The World $10.5 Trillion Annually By 2025. Article. Available at: https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021 [Accessed 2021-04-13].