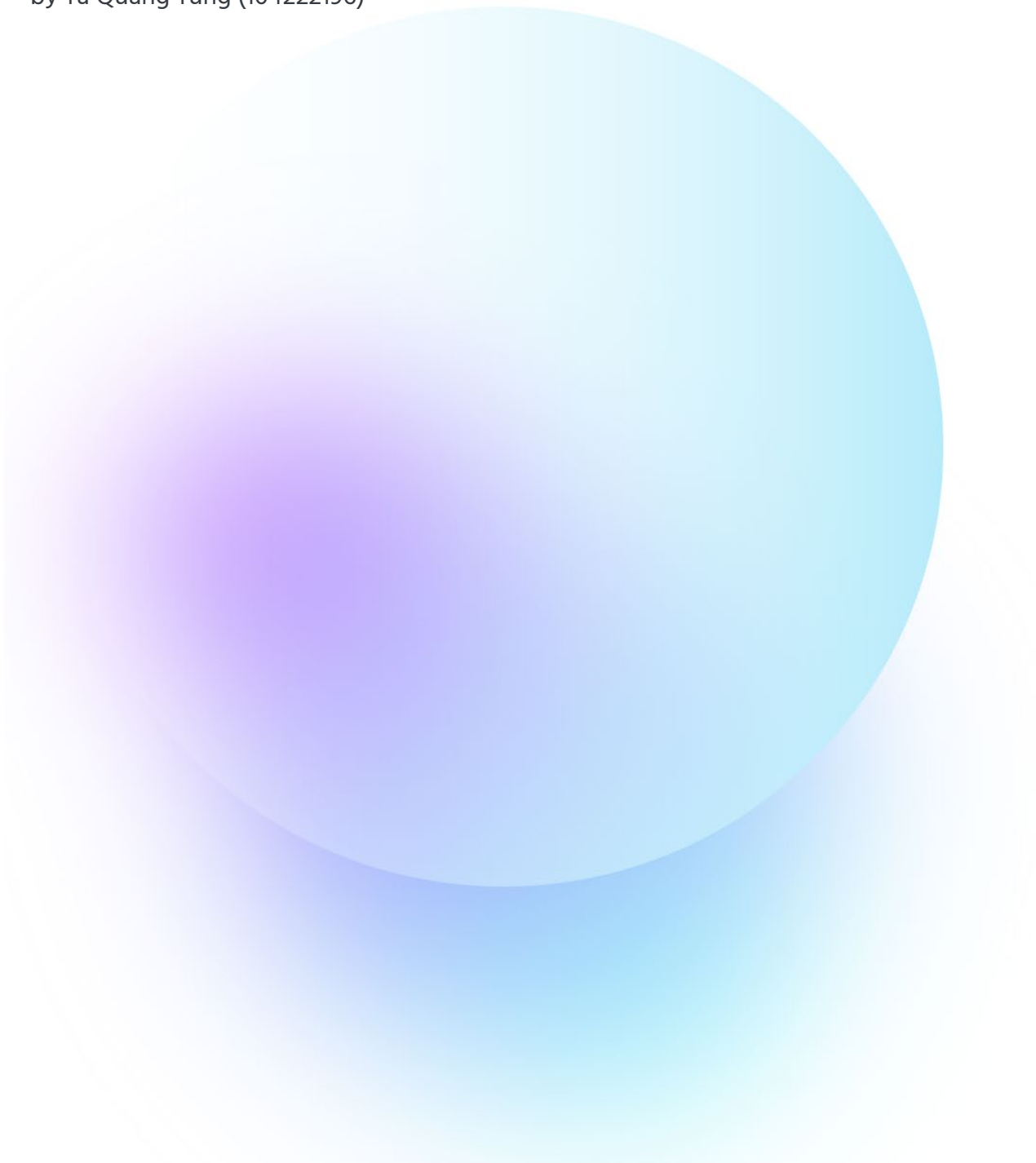


# **COS10026 Assignment 2**

## **Server-Side Programming Report**

by Ta Quang Tung (104222196)



## Introduction

This report presents the main features and implementation details of our jewelry website as part of Assignment 2 of unit COS10026 – Computing Technology Inquiry Project. This website is a collaborative effort between me and my team the Maverick Mates, which also includes Nguyen Quang Huy, Nguyen Thanh Trung, Nguyen Tran Quang Minh, and Pham Hung Manh. This project is a direct continuation of the static website we built for Assignment 1, but with more advanced features made possible by server-side programming with PHP and MySQL.

The report contains five sections. Section 1 describes the structure of the new website. Section 2 illustrates some key changes compared to the same static website of Assignment 1. Section 3 presents our server-side programming philosophies. Section 4 dives into the technical details of the site, including how we defined the database tables and used PHP for communicating between web pages. Section 5 details my contributions to the project and offers some reflection.

## Section 1 – Website structure

Our new website still retains the 5 basic pages made for Assignment 1, which are index.html, product.html, enquire.html (renamed to payment), about.html, and enhancements.html. However, one key difference is that the file extension has been changed from .html to .php to enable more complex features. We have also extracted the common header navigation bar and footer into separate includable files, header.inc and footer.inc, respectively, and included them in our pages.

To accommodate the requirements of Assignment 2, we have added many more pages. Many of these pages can be grouped into particular tasks, which are:

- Handling user orders: process\_order.php, fix\_order.php, receipt.php – These scripts work in conjunction with payment.php to save users' orders into the database.
- Allowing management: manager.php, edit\_order.php, delete\_order.php – These scripts allow management personnel to view, update the status, and remove orders from the database.
- Allowing user authentication (enhancement 1): register.php, login.php, change\_password.php, profile.php, process\_user\_register.php, process\_user\_login.php, process\_user\_logout.php, process\_change\_password.php – These scripts let users create accounts to see their shopping activities.
- Allowing management authentication (enhancement 2): manager\_register.php, manager\_login.php, process\_manager\_register.php, process\_manager\_login.php, process\_manager\_logout.php – These scripts provide security measures to restrict access to the management page.

Other miscellaneous pages include enhancements2.php, which describes the two enhancements we made for this assignment, and settings.php, which stores the credentials needed to connect to the database server.

As with our previous assignment, many of these pages are linked together with a common header navigation bar and footer.

## **Section 2 – Features and enhancements compared to Assignment 1**

For this assignment, the pages `index.php`, `product.php`, `about.php`, and `enhancements.php` remain almost the same as in assignment 1, with the addition of a few minor changes to make the content more relevant.

The page `enquire.html` (now `payment.php`) receives the biggest overhaul. Users can now order products through this page, which requires the addition of many input fields related to product, delivery, and payment information. The form is posted to `process_order.php`, where user inputs will be validated. If they are all valid, an order is created in the database and the user receives a receipt generated by `receipt.php`. Otherwise, they are taken to `fix_order.php` to correct the inputs and resubmit.

A new page, `manager.php` has been created for management purposes. This page displays orders stored in the database and offers filters such as first name, product name, cost, and status. Managers can also change the status of an order by a form that submits to `edit_order.php` and delete an order by a form that submits to `delete_order.php`.

As part of the enhancements, our website also offers user authentication. Users can choose to create an account that contains their personal and address information as well as information about their past orders. Having a user account auto-completes several information fields in the `payment.php` form, saving the user time. Users can see their personal information, purchase history, and favorite products on a dedicated `profile.php` page. They also have the option to change passwords.

We also provide authentication for managers. The `manager.php` page is locked by default and can only be accessed with a manager account. Logging in gives the manager a 30-minute session during which they can view and modify the manager page. As a weak security measure, we require an organizational security code which is “TTHMM” when creating an account to prevent access by outsiders.



### Order Form

**General Information**

First name:  
Tung

Last name:  
Ta

Email:  
tungnut@gmail.com

Phone number:  
1234567890

**Address**

Street Address:  
Thuy Khue

Suburb/Town:  
Hanoi

State:  
NSW

Postcode:  
1122

**Order**

Product:  
Please select product you want to purchase

*The payment.php page. Several fields have been auto-completed thanks to user authentication.*



### Register

Create an account to keep track of your purchase history and see your favorite purchases!

**Account information**

First name

Last name

Email

Phone  
E.g. 0123456789

Password

Confirm password

**Address information**

Street address

Suburb/town


State  
Please select

Postcode  
XXXX

Register

Already have an account? [Log in](#)

*User registration interface.*

[Home](#) [Products](#) [Purchase](#) [Manage](#) [About](#) [Enhancements I](#) [Enhancements II](#) [Profile](#) [Log out](#)

### Tung Ta's Profile

#### Your information

Email: tungnut@gmail.com  
Phone: 1234567890  
Address: Thuy Khue, Hanoi, NSW, 1122  
[Log out](#)

#### Purchase history

Your most recent purchases will appear here.

**\$50400**

**Maverick Mates Double Bracelet (gold, gold, diamond) x12**

Order ID: 2

Status: **FULFILLED**

Ordered at: Apr 10 2023 13:26:12 (UTC Time)

Ordered as: Tung Ta

Delivery address: Thuy Khue, Hanoi, NSW, 1122

#### Your top items

**#1 - Maverick Mates Double Bracelet (gold, gold, diamond)**

You have purchased 1 time(s).


Maverick Mates - unafraid of being yourself.

Sitemap  
[Home](#) [Product](#) [Enquiry](#) [Manage](#) [About](#) [Enhancements I](#) [Enhancements II](#)

Connect  
[Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

© Copyright Maverick Mates 2023

*The user's profile page. Here they can see their personal information as well as purchase history.*

[Home](#) [Products](#) [Purchase](#) [Manage](#) [About](#) [Enhancements I](#) [Enhancements II](#) [Profile](#) [Log out](#)

### Manage orders

You are logged in as admin. You will be automatically logged out in 29 minute(s) and 54 second(s). [Log out.](#)

First name:  Sort order cost:  Product:  Status:

ID	Date of order	Product	Total cost	First name	Last name	Status	Change Status	Delete
31	2023-04-09 10:46:14	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
32	2023-04-09 10:46:16	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
33	2023-04-09 10:46:18	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
34	2023-04-09 10:46:21	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
29	2023-04-09 10:45:30	Maverick Mates Gemstone Pendant (gold, gold, diamond)	16200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
38	2023-04-09 13:06:34	Maverick Mates Knot Ring (gold, gold, citrine)	9298	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
35	2023-04-09 13:06:11	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
36	2023-04-09 13:06:16	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
37	2023-04-09 13:06:19	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>

*The manager.php page.*

## Section 3 – Server-side design philosophies

Many of our pages require accessing and reading from a common database table named “orders”. As such, before diving straight into the PHP, we needed to define a common table schema that would satisfy the needs of all the different pages. We choose our table schema as defined by the following query:

```
CREATE TABLE orders (  
    order_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(25) NOT NULL,  
    last_name VARCHAR(25) NOT NULL,  
    email TEXT NOT NULL,  
    phone VARCHAR(10) NOT NULL,  
    product_name TEXT NOT NULL,  
    product_amount INT NOT NULL,  
    order_cost FLOAT NOT NULL,  
    card_name ENUM('American Express', 'Mastercard', 'Visa') NOT NULL,  
    card_number VARCHAR(16) NOT NULL,  
    card_owner TEXT NOT NULL,  
    card_expiry VARCHAR(5) NOT NULL,  
    cvv VARCHAR(3) NOT NULL,  
    street_address VARCHAR(40) NOT NULL,  
    suburb VARCHAR(20) NOT NULL,  
    state ENUM('ACT', 'NSW', 'NT', 'QLD', 'SA', 'TAS', 'VIC', 'WA') NOT NULL,  
    postcode VARCHAR(4) NOT NULL,  
    order_time DATETIME NOT NULL,  
    order_status ENUM('ARCHIVED', 'FULFILLED', 'PAID', 'PENDING') NOT NULL  
);
```

One of our enhancements requires associating an order with a user. We decided not to add a foreign key field referencing a user directly into the orders table to minimize impact should we make any changes. Therefore, we had to create a separate table containing foreign keys that reference users and orders. The schema for it is as follows:

```
CREATE TABLE user_order (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    order_id INT NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE  
);
```

Another feature of our project is the heavy use of PHP session variables to share data between pages. For instance, the user's authentication data such as user ID needs to be passed between pages to control access. Since all variables are stored within the same superglobal `$_SESSION`, we needed to conform to a strict naming convention to prevent clashing. As an example, all variables related to user authentication are marked with the prefix "user\_", while all variables related to management authentication are marked with "manager\_".

In short, our server-side design philosophies are: consistent table schema, table separation for minimal impact, and non-clashing naming convention.



## Section 4 – Implementation

### I – Additional tables

This assignment requires the creation of MySQL tables to store user-generated data. To fulfill the basic requirements as well as our enhancements, we need to add two more tables, including users and managers. The schemas for these tables are defined by the following queries:

```
CREATE TABLE users (  
    user_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(25) NOT NULL,  
    last_name VARCHAR(25) NOT NULL,  
    email TEXT NOT NULL,  
    phone VARCHAR(10) NOT NULL,  
    street_address VARCHAR(40) NOT NULL,  
    suburb VARCHAR(20) NOT NULL,  
    state ENUM('ACT', 'NSW', 'NT', 'QLD', 'SA', 'TAS', 'VIC', 'WA') NOT NULL,  
    postcode VARCHAR(4) NOT NULL,  
    password TEXT NOT NULL  
);  
  
CREATE TABLE managers (  
    manager_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    username TEXT NOT NULL,  
    password TEXT NOT NULL  
);
```

### II – Getting orders associated with a user using INNER JOINS

Users are linked to their orders via a user\_order table containing only foreign keys, as shown by the schema. Therefore, a simple select statement is not enough to obtain all the information. We needed to incorporate INNER JOIN statements to connect the separate tables.

To select the 25 most recent orders made by a user of ID user\_id, we used the following query:

```
SELECT orders.order_id, orders.first_name, orders.last_name, product_name,  
product_amount, order_cost, orders.street_address, orders.suburb,  
orders.state, orders.postcode, order_time, order_status FROM user_order  
INNER JOIN users ON user_order.user_id = users.user_id  
INNER JOIN orders ON user_order.order_id = orders.order_id  
WHERE users.user_id = '$user_id'  
ORDER BY order_time DESC  
LIMIT 25;
```

To select the top 3 products ordered by a user of ID user\_id, we used the following query:

```
SELECT product_name, COUNT(orders.order_id) AS count FROM user_order  
INNER JOIN users ON user_order.user_id = users.user_id  
INNER JOIN orders ON user_order.order_id = orders.order_id  
WHERE users.user_id = '$user_id'  
GROUP BY product_name  
ORDER BY count DESC  
LIMIT 3;
```

These statements are run by the profile.php page, where user purchase data is fetched.

### III – Access control and redirection

Unlike the previous assignment where every page is accessible at all times, in this assignment, we need to restrict direct access to many pages. For instance, users should not be able to directly access process\_order.php or access profile.php without authentication. To achieve this, we made extensive use of session variables, which is a secure way to pass data around pages. Access to process\_order.php is provided only if the user has a special token added to the session superglobal by payment.php or fix\_order.php. Meanwhile, access to profile.php is provided only if the user\_id session variable is present, indicating that the user has indeed logged in.

When users enter a page where they should not be, they will be redirected to a public page such as index.php. Redirection is achieved with the function header("Location: ...");

### IV – Overcoming the obstacle of no JavaScript

As part of the requirements of the manager.php page, we have to somehow enable users to communicate a message to the server to update or delete an order. With JavaScript, this might be achieved by making a request to the server using an API such as fetch(). However, since JavaScript is not allowed in this assignment, we figured out an alternative method involving forms and hidden inputs. Each update status or delete button of an order will be contained in a form along with a hidden input featuring the ID of that order. When the processing file on the server (delete\_order.php / edit\_order.php) receives this form request, it will read the attached ID and delete/update accordingly.

## V – Setting a time limit for a management session

For our management authentication enhancement, we decided that it would be appropriate to limit a management session to 30 minutes for security reasons. This means that a manager has to login again every 30 minutes, otherwise the manager.php page becomes inaccessible. To achieve this, whenever a user logs in as a manager, a timestamp is saved in the session. Every time the manager.php page is requested, the server calculates if the user has exceeded the 30-minute limit. If yes, it unsets the manager\_id session variable, effectively logging out the manager. The code for this is as follows,

in process\_manager\_login.php:

```
<?php
    session_start();
    // Set the login timestamp to current time.
    $_SESSION["manager_login_time"] = time();
?>
```

in manager.php:

```
<?php
    session_start();
    // This page is only accessible to authenticated managers. Authentication
    for management lasts 30 minutes only.
    // If the logged-in manager has exceeded this time limit, log them out
    automatically.
    if (isset($_SESSION["manager_login_time"]) && time() -
    $_SESSION["manager_login_time"] > 1800) {
        unset($_SESSION["manager_id"], $_SESSION["manager_username"],
        $_SESSION["manager_login_time"]);
    }
```

```
// If the manager is not logged in, as indicated by the absence of the
manager_id session variable, take
// them to the login page.
if (!isset($_SESSION["manager_id"])) {
    header("Location: manager_login.php");
    die();
}
?>
```

tes

### Manage orders

You are logged in as admin. You will be automatically logged out in 29 minute(s) and 54 second(s). [Log out](#).

First name:  Sort order cost:  Product:  Status:

ID	Date of order	Product	Total cost	First name	Last name	Status	Change Status	Delete
31	2023-04-09 10:46:14	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
32	2023-04-09 10:46:16	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
33	2023-04-09 10:46:18	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
34	2023-04-09 10:46:21	Maverick Mates Gemstone Pendant (gold, gold, diamond)	97200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
29	2023-04-09 10:45:30	Maverick Mates Gemstone Pendant (gold, gold, diamond)	16200	Quang	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
38	2023-04-09 13:06:34	Maverick Mates Knot Ring (gold, gold, citrine)	9298	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
35	2023-04-09 13:06:11	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
36	2023-04-09 13:06:16	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>
37	2023-04-09 13:06:19	Maverick Mates Double Bracelet (gold, gold, diamond)	4200	Tung	Ta	PENDING	<input type="button" value="Update status"/>	<input type="button" value="Cancel order"/>

*The manager.php page. Notice a time limit is given to the user at the top.*

## Section 5 – Contributions and reflections

My contributions to this assignment include defining the table schemas, working on the two enhancements, and putting my team's work together.

Since my work entails much experimentation on the tables, I had to ensure that any changes I made would not affect the work of my teammates. As such, I decided to not complicate the shared orders table with foreign keys and instead create a separate user-order table for my own development.

My work requires the extensive use of session variables, so another concern I had was to avoid interfering with the session variables defined by my teammates' scripts. I mitigated this risk by prefixing all my session variables to uniquely identify them and unsetting only specific variables at any given time instead of destroying everything.

As part of my job assembling the work of my teammates, I also had to make sure that their code is properly commented with documentation and header comments. This allows me to learn their different approaches to the problem defined by the requirements, and in the process, understand how the entire project works as a whole.

This project has familiarized me with the process of server-side development. Key lessons I have learned from this project are to (1) have a clearly defined and agreed-upon table schema for the data and (2) know to select the appropriate form of data passing between pages (through query strings, session variables, or forms). This assignment has also given me the opportunity to implement a basic security system for my website via user and manager authentication. However, this system has a major unaddressed flaw. Our website gives users the choice to change their password, but all that is required to request a password change is the email of the account. Thus, anyone could hijack an account by obtaining its email address. A real-world application would implement security measures such as phone or email verification to prevent account hijacking. However, since this assignment is for demonstrative purposes only, I have decided to skip this complex part.

## Summary

This report describes the various features and technical server-side aspects of our website as part of Assignment 2 of unit COS10026. Points covered include the new website structure, features, design principles, and code implementations as well as my contributions. The final section offers a few points of reflection and points out a security weakness that can be improved in the future.