



WENZHOU-KEAN  
UNIVERSITY

## Final Project

CPS3500: Programming World Wide Web Server

Project Name:	Campus Map
Student Name:	Chun Chen (Dave)
Student ID:	1234930
Lecture:	Dr. Hemn Barzan Abdalla

Spring 2025

## Table of Content

1. Introduction
2. Proposed problem
3. Proposed solution
4. Conclusion
5. References

## 1. Introduction

This website provides a visual and interactive campus navigation tool, featuring a login and registration page, a main map interface, and detailed pages for selected buildings.

On the main map page, users can quickly identify building names, relative positions, and images through interactive SVG elements.

Additionally, users can access detailed building pages by clicking on the corresponding areas in the SVG map or by using the navigation bar.

## 2. Proposed Problem

The website requires a functional database to support user login and registration for accessing the campus map system.

In addition, the main map interface needs to offer sufficient interactivity and ease of use, allowing users to quickly understand building positions and names.

To enhance the overall experience, each building should also have a dedicated page that provides detailed information and images.

## 3. Proposed Solution

The whole System uses the following structure:

CampusMap(Root)-----index.html(for sign in and signup)

    |---campusMap.html(for the main map)

    |---BusinessFaculty.html(detail of CBPM)

    |---ComprehensiveStadium.html(detail of the Comprehensive Stadium)

    |---GHKBuilding(detail of the GHKBuilding)

    |---db.php(get user data from database)

    |---signup.php(insert new user data through db.php)

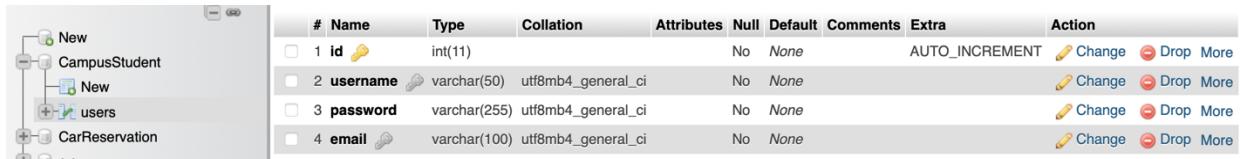
    |---login.php(use the user data from db.php to check in)

    |---logout.php clear the login status)

    |---imgs(contains images in the building page, and some background images)

|---map(contains images needed in the main map, and some background images)

|---Style.css(contains all style we need)



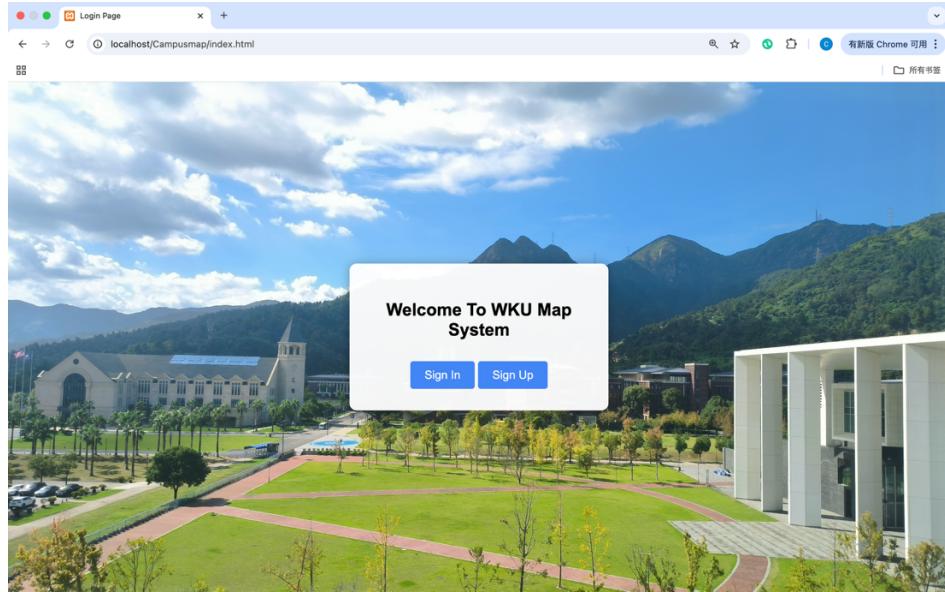
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>id</b>	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	<b>username</b>	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
3	<b>password</b>	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
4	<b>email</b>	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More

(Fig.1)

This project uses a MySQL database. The database is named **CampusStudent**, and it contains a table named **users**, which includes 4 fields: id, username, password, and email. (Fig.1)

### The Login and Signup page(index.html):

The start page look like below(Fig.2):



(Fig.2)

If we click Sign in or Sign up, this form will change to a sign-in form(id = signin-form) or a sign-up form(id = signup-form)(Fig3, 4, 5).

```
<form id="signin-form" action="login.php" method="POST" style="display: none;">
<form id="signup-form" action="signup.php" method="POST" style="display: none;">
```

(Fig.3)

```

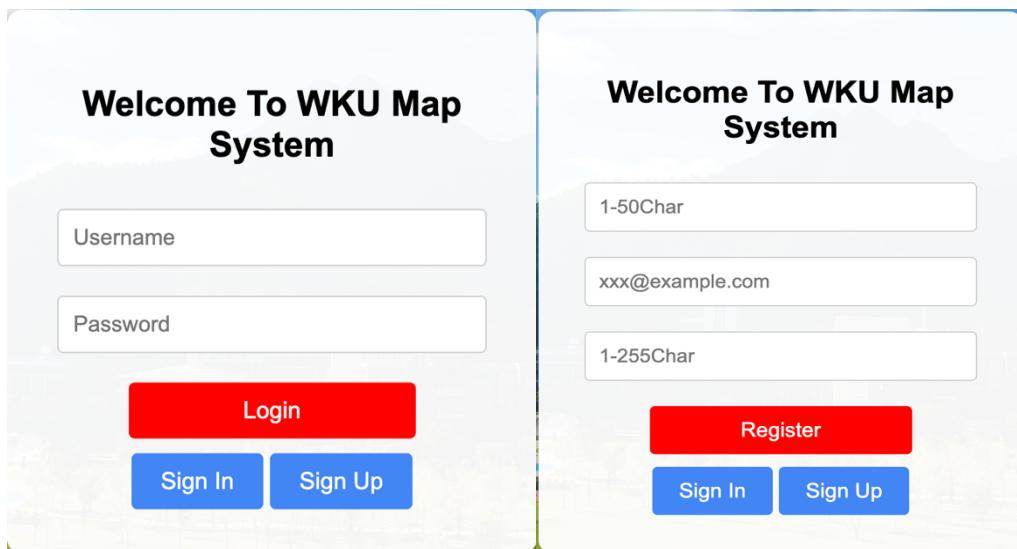
<div>
  <button onclick="showForm('signin')>Sign In</button>
  <button onclick="showForm('signup')>Sign Up</button>
</div>

</div>
</body>

<script>
  function showForm(type) {
    document.getElementById("signin-form").style.display = type === "signin" ? "block" : "none";
    document.getElementById("signup-form").style.display = type === "signup" ? "block" : "none";
  }
</script>

```

(Fig.4)



(Fig.5) left is Sign In, right is Sign Up

The data we be submitted to login.php or signup.php(Fig.3, 6),

```
<button type="submit" class="LR-button">Login</button>
```

```
<button type="submit" class="LR-button">Register</button>
```

(Fig.6)

The login.php file uses the PDO object from db.php to retrieve user information from the database. If the user exists, it compares the input password with the one stored in the database. The signup.php file uses an SQL INSERT statement to add new user information (username, email, password) into the users table via the same database connection provided by db.php.(Fig. 7, 8, and 9)

```
try {
    $pdo = new PDO($dsn, $user, $pass);
```

(Fig.7)db.php

```
if ($user && $password === $user['password']) {
    $_SESSION['username'] = $username;
    header("Location: campusMap.html");
    exit();
} else {
```

(Fig.8)login.php

```
$stmt = $pdo->prepare("INSERT INTO users (username, email, password) VALUES (?, ?, ?)");
$stmt->execute([$username, $email, $password]);
```

(Fig9)signup.php

### The Main Map page(CampusMap.html):

use figma form **SVG** picture with id(Fig.11) and translate into html code(Fig.10) and use in(Fig.12):

```
<path class="map-area" d="M305 426.5V397H323V402H339.5V392H347V397H350.5V419.5H358V426.5H305Z"
| fill="#FFF59D" fill-opacity="0.7" id="StudentLearning&ActivityCenter" stroke="black" />
```

(Fig.10)

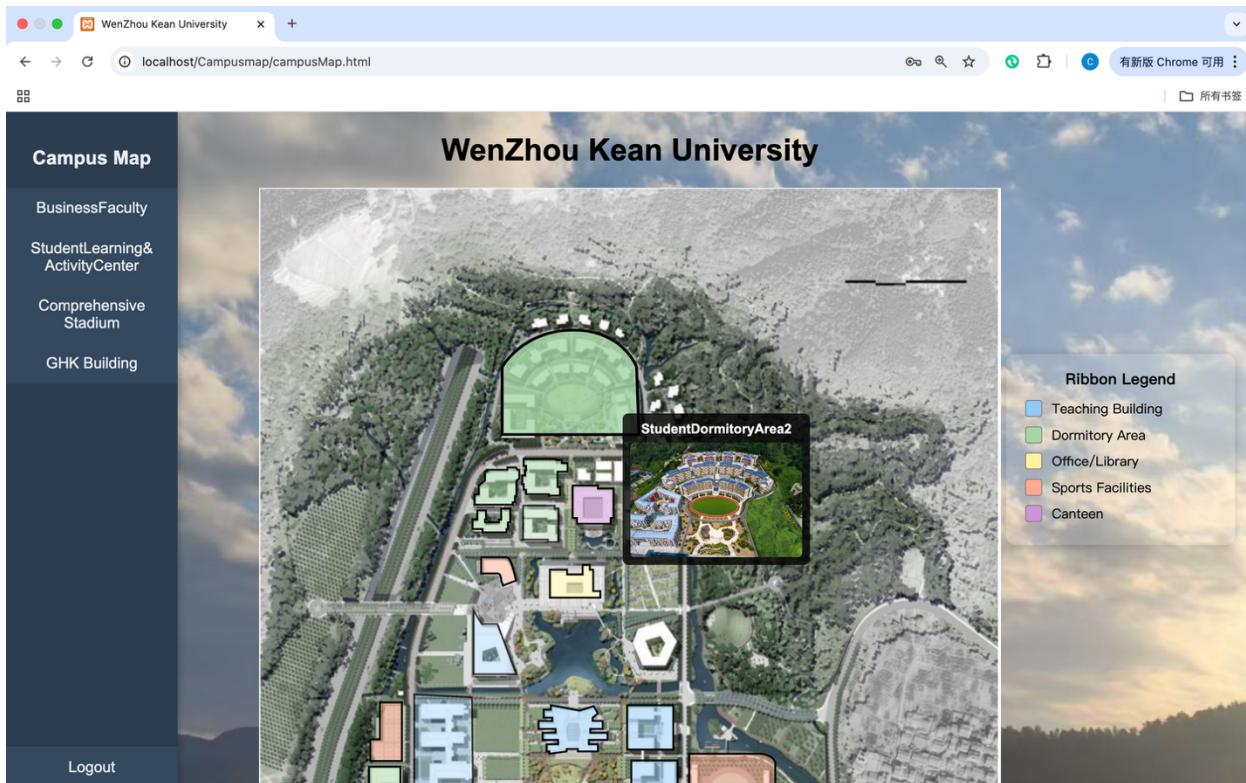
d="\*\*\*\*" describes the path direction and length, fill="\*\*\*\*" describes the internal color, fill-opacity"\*\*\*\*" describes the transparency, and stroke is the border color.

We also use the code below to ensure that the browser can correctly interpret and render SVG

31 | | | <svg fill="none" height="908" viewBox="0 0 778 902" width="778" xmlns="http://www.w3.org/2000/svg">



(Fig.11)figma SVG



(Fig.12)

We add eventListener to achieve interactive functions

```
<div id="tooltip">
  <div id="tooltip-name"></div>
  <img id="tooltip-img" src="" alt="BuildingImage" />
</div>

const tooltip = document.getElementById("tooltip");
const tooltipName = document.getElementById("tooltip-name");
const tooltipImg = document.getElementById("tooltip-img");

areas.forEach(area => {
  area.addEventListener("mousemove", (e) => {
    const id = area.id;
    tooltip.style.display = "block";
    tooltip.style.left = e.pageX + 10 + "px";
    tooltip.style.top = e.pageY + 10 + "px";

    tooltipName.innerText = id;

    if (buildingImages[id]) {
      tooltipImg.src = buildingImages[id];
      tooltipImg.style.display = "block";
    } else {
      tooltipImg.style.display = "none";
    }
  });

  area.addEventListener("mouseleave", () => {
    tooltip.style.display = "none";
  });

  area.addEventListener('click', () => {
    const id = area.id;
    const url = `${id}.html`;

    fetch(url, { method: 'HEAD' })
      .then(response => {
        if (response.ok) {
          window.location.href = url;
        } else {
          alert(`Page ${id} Not Defined, Please Wait!!`);
        }
      })
      .catch(() => {
        alert(`Unable to access the webpage ${url}`);
      });
  });
});
```

(Fig.13)

If the pointer passes through the SVG with the corresponding id, the corresponding image and id will be displayed.(Fig13)

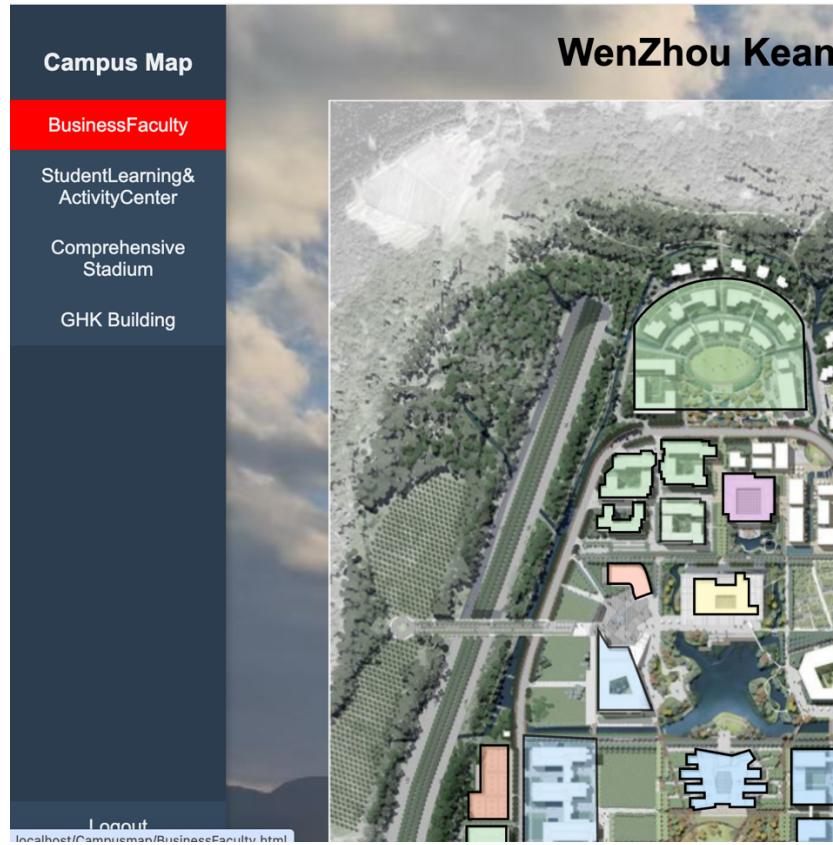


(Fig.14)

And if we click the SVG, it will jump to the corresponding building page see below(GHKBuilding.html). All building pages use the same framework and CSS:

A screenshot of a web browser window. The title bar says "GHK Building". The address bar shows "localhost/Campusmap/GHKBuilding.html". The main content area has a dark header with "GHK Building". Below the header is a text block about the building, followed by three images: a sunset over the building, an interior classroom, and a close-up of a wall. The browser interface includes standard navigation buttons and a toolbar.

We can also jump to the same page by clicking the building name corresponding to the sidebar on the left side of the webpage.(Fig. 15)



(Fig.15)sidebar

We also added a Ribbon Legend to increase visualization, and matched the corresponding functional areas with the colors on the graph.(Fig.16)

```
<div class="legend-box">
  <h3>Ribbon Legend</h3>
  <ol>
    <li><span class="legend-color" style="background-color: #90CAF9;"></span> Teaching Building</li>
    <li><span class="legend-color" style="background-color: #A5D6A7;"></span> Dormitory Area</li>
    <li><span class="legend-color" style="background-color: #FFF59D;"></span> Office/Library</li>
    <li><span class="legend-color" style="background-color: #FFAB91;"></span> Sports Facilities</li>
    <li><span class="legend-color" style="background-color: #CE93D8;"></span> Canteen</li>
  </ol>
</div>
```

(Fig.16)

**About Style.css:**

```
1  /* for index */
2  .login-container {
|    ...
|
24 /* for signin and sigup button */
25 .login-container button {
```

(Fig.17)

For every CSS block(Fig.17) we have already added comments to describe function, For example:

```
312 /* four building page use same style */
313 .container {
314     max-width: 900px;
315     margin: 40px auto;
316     background: white;
317     padding: 30px;
318     border-radius: 10px;
319     box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
320     box-sizing: border-box;
321 }
322
323 .text-align-center {
324     text-align: center;
325 }
326
327 .text-align-center img {
328     vertical-align: middle;
329     width: 100%;
330     height: auto;
331 }
```

(Fig. 18)

This block(Fig.18) describes four Building pages that use the same framework(Fig.19)

```

<div class="container">
  <p>
    The architectural design of Wenzhou-Kean University campus seeks diverse changes, providing great space for innovative campus design. The campus does not insist on unified coordination, but encourages buildings to communicate and dialogue in diversity, and find ways and methods of balance and coordination.
    <br>

    Therefore, unlike other buildings on campus, the business school adds a bit of personality and vitality to the Wenzhou-Kean University campus with its modern design style and bright color matching. At the same time, the business school's facade color selection and large steps and other design elements echo the existing campus buildings, forming communication and dialogue in diversified development.
  </p>
  <p class="text-align-center">Exterior</p>
  <p class="text-align-center">Lecture Hall</p>
  <p class="text-align-center">Office</p>
</div>

```

(Fig. 19)

## 4. Conclusion

This project successfully implements an interactive campus map system that combines user authentication and dynamic visual navigation.

Users can sign in or sign up through a unified login interface, which is connected to a MySQL database using PHP (via db.php) to store and verify account data securely.

Once authenticated, users are redirected to the main campus map page, where interactive SVG elements allow them to explore building names, view images, and access dedicated building information pages with a single click.

Each building page follows a consistent design framework and shares a unified style defined in style.css.

The sidebar navigation and color-coded legend further enhance usability and provide intuitive access to specific areas of the campus.

Through this system, the project meets its primary goals: improving navigation clarity, enabling easy user access control, and presenting detailed building information in a user-friendly format. It demonstrates the integration of front-end visualization with back-end data handling in a clean and modular structure.

## 5. References

- Figma was used to design and export the interactive SVG image.
- The standard from <http://www.w3.org/2000/svg> was used to ensure correct parsing and rendering of SVG elements in the browser.
- A users table was defined using MySQL to support login and registration.
- Visual Studio Code was used as the primary code editor and development environment.