

《数据库系统》课程设计报告

题目	姓名	学号	班级	提交时间
邮局订报管理子系统	张美璇	20337266	计科二班	2023.1.2

系统功能要求

设计本系统模拟客户在邮局订购报纸的管理内容，包括查询报纸、订报纸、开票、付钱结算、订购后的查询、统计等的处理情况。简化的系统需要管理的情况如下：

- (1) 可随时查询出可订购报纸的详细情况，如报纸编号 (pno) 、报纸名称 (pna) 、报纸单价 (ppr) 、版面规格 (psi) 、出版单位 (pdw) 等，这样便于客户选订。
- (2) 客户查询报纸情况后即可订购所需报纸，可订购多种报纸，每种报纸可订若干份，交清所需金额后，就算订购处理完成。
- (3) 为便于邮局投递报纸，客户需写明如下信息：客户姓名 (gna) 、电话 (gte) 、地址 (gad) 及邮政编码 (gpo) ,邮局将即时为每一个客户编制唯一代码 (gno) 。
- (4) 邮局对每种报纸订购人数不限，每个客户可多次订购报纸，所订报纸也可重复。

根据以上信息完成如下要求：

- (1) 请认真做系统需求分析，设计出反映本系统的ER图（需求分析、概念设计）。
- (2) 写出相应你设计的ER图的关系模式，根据设计需要也可增加关系模式，并找出各关系模式的关键字（逻辑设计）。
- (3) 在你设计的关系模式基础上利用VB、C#或Java+MySQL（或其他开发设计平台）开发设计该子系统，要求子系统能完成如下功能要求（物理设计、设施与试运行）：
 - ①在MySQL中建立各关系模式对应的库表，并确定索引等；
 - ②能对各库表进行输入、修改、删除、添加、查询、打印等基本操作；
 - ③能根据订报要求订购各报纸，并完成一次订购任务后汇总总金额，模拟付钱、开票操作；
 - ④能明细查询某客户的订报情况及某报纸的订出情况；
 - ⑤能统计出某报纸的总订数量与总金额及某客户订购报纸种数、报纸份数与总订购金额等；
 - ⑥其他你认为子系统应有的查询、统计功能；
- (4) 子系统设计完成后请书写课程设计报告，设计报告要围绕数据库应用系统开发设计的步骤来考虑书写，力求清晰流畅。最后根据所设计子系统与书写报告（报告按数据库开发设计6个步骤的顺序逐个说明表达，并说明课程设计体会等）好坏评定成绩。

一、开发环境与开发工具

操作系统: Windows 11

开发语言: html + Javascript + css + php

web服务器: Apache 2.4.54 win64 VS17

数据库: Mysql 8.0.31-1.el8

数据库服务器: Docker version 20.10.17

二、系统需求分析

2.1 系统概述

本系统是一个邮局订报管理系统，主要功能是实现报纸的订购、查询、统计等功能。本系统的用户包含两类，一类是报社的客户，另一类是报社的工作人员。

报社的客户可以通过本系统进行登录和注册，并进行报纸的订购、查询、统计等操作。

报社的工作人员可以通过本系统对系统数据进行管理，包括对报纸、客户、订报信息等数据进行增加、删除、修改、查询等操作。

2.2 系统功能

面向客户，本系统的主要功能有：

1. 客户注册。客户可以通过本系统进行注册，注册时需填写个人信息（姓名、电话、邮箱、地址、邮编和账号密码）。
2. 客户登录。在本系统注册过账号的客户在登录页面填写个人的电话和密码，经过验证后能够登录本系统。
3. 查看报纸信息。客户在登录后能够查看所有报纸信息（包括报纸的名称、价格、出版周期、出版社和报纸规格）。
4. 订购报纸。客户在登录后能够挑选报纸进行订购，订购后的报纸信息会保存在数据库中。
5. 查看个人信息。客户在登录后能够查看在注册时填写的个人信息，如果需要修改可联系管理员。
6. 查看个人的历史订单。客户在登录后能够查看自己的历史订单信息（包括订单号、报纸名称、订购数量、订购周期、总计价格和订购日期）。
7. 修改登录密码。
8. 退出登录。

面向管理员，本系统的主要功能有：

1. 管理员登录。管理员在登录页面填写个人的账号和密码，经过验证后能够登录本系统。
2. 管理报纸信息。管理员在登录后能够对报纸信息进行增加、删除、修改、查询。
3. 管理订单信息。管理员在登录后能够对订单信息进行增加、删除、修改、查询。
4. 管理客户信息。管理员在登录后能够对客户信息进行增加、删除、修改、查询。
5. 退出登录。

2.3 系统数据字典

2.3.1 Guest

字段名	数据类型	默认值	允许非空	自动递增	备注
id	varchar(8)	/	no	yes	客户编号, 唯一
password	varchar(60)	/	no	no	客户密码
name	varchar(30)	/	no	no	客户姓名
phone	varchar(20)	/	no	no	客户电话
email	varchar(20)	/	no	no	客户邮箱
address	varchar(100)	/	no	no	客户地址
postcode	varchar(6)	/	no	no	客户邮编

2.3.2 Newspaper

字段名	数据类型	默认值	允许非空	自动递增	备注
id	varchar(8)	/	no	yes	报纸编号, 唯一
name	varchar(30)	/	no	no	报纸名称
price	double	/	no	no	报纸价格
frequency	varchar(10)	/	no	no	报纸发行频率 (以30天算)
publisher	varchar(30)	/	no	no	报纸出版社
height	int	/	no	no	报纸高度(以mm为单位)
width	int	/	no	no	报纸宽度(以mm为单位)

2.3.3 Order

(此处设定的order是单次订购中一种报纸的订单, 单次订购单的总订单为Bill, Bill包含一个或多个Order)

字段名	数据类型	默认值	允许非空	自动递增	备注
id	varchar(8)	/	no	yes	订单编号, 唯一
guest_id	varchar(8)	/	no	no	客户编号, 包含到Guest的外码约束

字段名	数据类型	默认值	允许非空	自动递增	备注
newspaper_id	varchar(8)	/	no	no	报纸编号, 包含到Newspaper的外码约束
number	int	/	no	no	订购数量
period	int	1	no	no	订购周期 (以30天算)

2.3.4 Bill

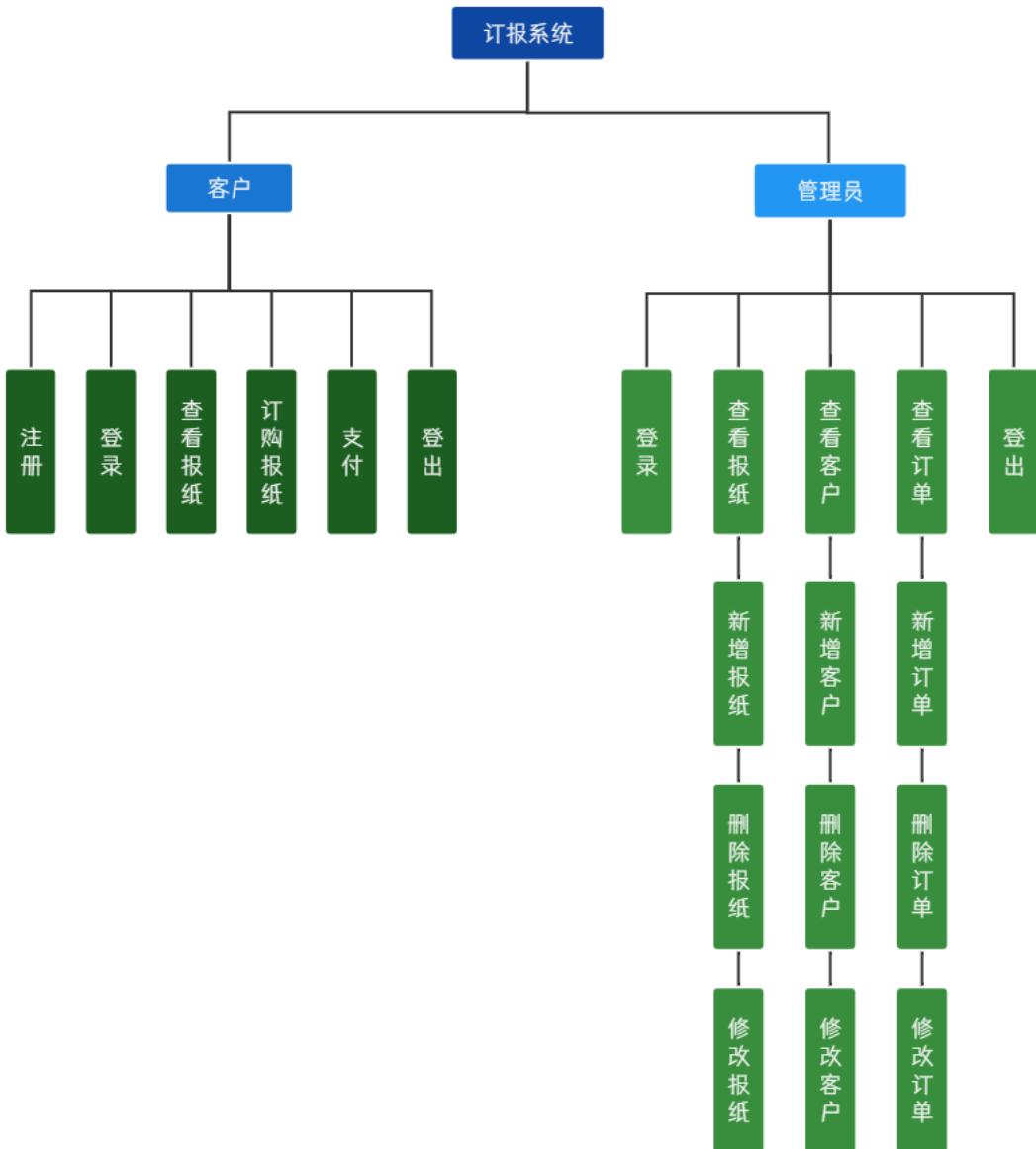
字段名	数据类型	默认值	允许非空	自动递增	备注
order_id	varchar(8)	/	no	no	订单编号, 包含到Order的外码约束
time	datetime	/	no	no	下单时间

2.3.5 Manager

字段名	数据类型	默认值	允许非空	自动递增	备注
id	varchar(8)	/	no	yes	管理员编号, 唯一
name	varchar(30)	/	no	no	管理员姓名
password	varchar(60)	/	no	no	管理员密码

三、功能需求分析

绘制系统功能模块图如下：



四、系统设计

4.1 数据概念结构设计

4.1.1 实体类型

在本系统中，实体类型有Guest、Newspaper、Manager；

Guest实体类型的属性有id、password、name、address、telephone、email、postcode；

Newspaper实体类型的属性有id、name、price、frequency、publisher、height、width；

Manager实体类型的属性有id、name、password；

4.1.2 关系类型

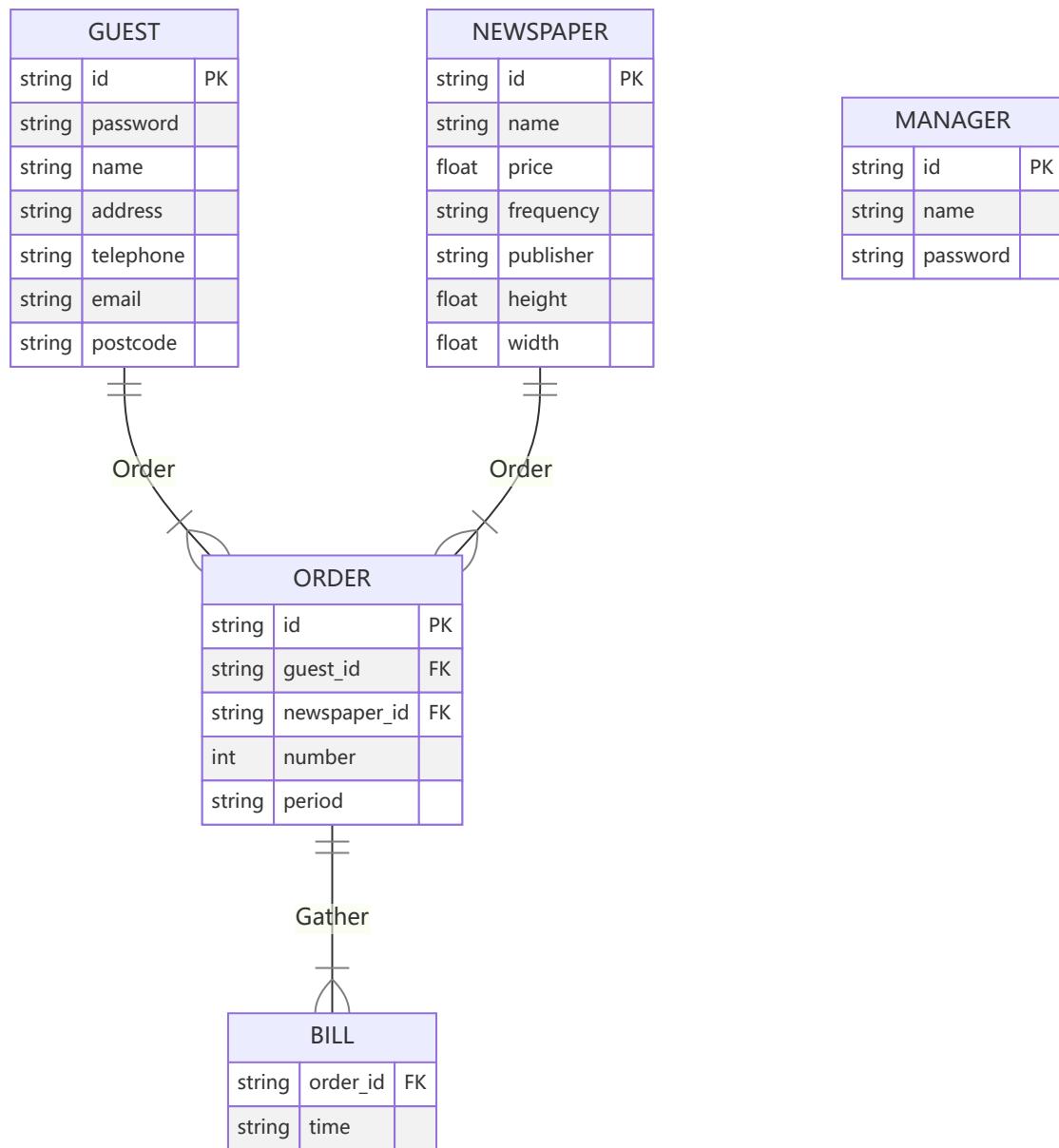
在本系统中，关系类型有Order、Gather；

Order关系类型是Guest订购Newspaper产生的关系，它的属性有id、guest_id、newspaper_id、number、period,其中guest_id和newspaper_id是外码，分别引用Guest和Newspaper实体类型的id属性；

Gather关系是对Order的一次订购汇总，它的属性有id、order_id、time,其中order_id是外码，引用Order实体类型的id属性；

4.1.3 ER图

根据上述实体类型和关系类型，绘制ER图如下：



4.2 数据关系模式设计

下面将上述ER图转换为关系模式：

首先将ER图中的实体类型转换为关系模式，得到如下关系模式：

Guest(id, password, name, address, telephone, email, postcode)
Newspaper(id, name, price, frequency, publisher, height, width)
Manager(id, name, password)
Order(id, guest_id, newspaper_id, number, period)
Bill(order_id, time)

(Bill是一个弱实体类型，它的主码是time和order_id，它的外码是order_id，它的主码和外码都引用Order实体类型的id属性。)

然后将ER图中的关系类型转换为关系模式：

Order(id, guest_id, newspaper_id, number, period)
Gather(id, order_id, time)

最终得到如下关系模式：

Guest(id, password, name, address, telephone, email, postcode)
Newspaper(id, name, price, frequency, publisher, height, width)
Order(id, guest_id, newspaper_id, number, period)
Bill(order_id, time)
Manager(id, name, password)

4.3 数据物理结构设计

在本系统中，数据物理结构设计主要是对关系模式的属性进行约束，以及对关系模式的属性进行索引。

4.3.1 约束

对Guest实体类型的属性进行约束，得到如下约束：

属性	是否为主码	外码约束
id	是	无
password	否	无
name	否	无
address	否	无
telephone	否	无
email	否	无
postcode	否	无

对Newspaper实体类型的属性进行约束，得到如下约束：

属性	是否为主码	外码约束
id	是	无
name	否	无

属性	是否为主码	外码约束
price	否	无
frequency	否	无
publisher	否	无
height	否	无
width	否	无

对Order实体类型的属性进行约束，得到如下约束：

属性	是否为主码	外码约束
id	是	无
guest_id	否	是，引用Guest的id属性
newspaper_id	否	是，引用Newspaper的id属性
number	否	无
period	否	无

对Bill实体类型的属性进行约束，得到如下约束：

属性	是否为主码	外码约束
order_id	否	是，引用Order的id属性
time	否	无

对Manager实体类型的属性进行约束，得到如下约束：

属性	是否为主码	外码约束
id	是	无
name	否	无

4.3.2 索引

由于本系统中的查询操作主要是对Guest、Newspaper、Order、Bill、Manager实体类型的属性进行查询，所以对这些实体类型的属性进行索引。在具体实现时，对Guest实体类型的id属性、Newspaper实体类型的id属性、Order实体类型的id属性、Bill实体类型的order_id属性、Manager实体类型的id属性进行索引。

4.4 数据库的实现

在实现数据库时，首先创建数据库，然后创建表，最后对表进行约束和索引。使用的SQL语句如下：

```
create database SubscriptionSystem;
use SubscriptionSystem;

create table Guest(
    gno varchar(8) not null ,
    gname varchar(30) not null ,
    gpassword varchar(20) not null ,
    gemail varchar(30),
    gaddress varchar(100) not null ,
    gtelephone varchar(20),
    gpostcode varchar(6) not null ,
    primary key (gno)
);

alter table Guest
modify gtelephone varchar(20) not null ;

alter table Guest
modify gemail varchar(20) not null ;

create table Paper(
    pno varchar(4) not null ,
    pname varchar(30) not null ,
    pprice decimal(5,2) not null ,
    pwidth int,
    pheight int,
    ppublisher varchar(30),
    primary key (pno)
);

alter table Paper
add column frequency int default 30 not null after pprice;

create table Orders(
    ono varchar(10) not null ,
    gno varchar(8) not null ,
    pno varchar(4) not null ,
    onumber int not null ,
    primary key (ono),
    foreign key (gno) references Guest(gno),
    foreign key (pno) references Paper(pno)
);

alter table Orders
add column period int default 1 not null after onumber;

create table Bill(
    btime datetime not null,
    ono varchar(10) not null ,
    primary key (btime,ono),
    foreign key (ono) references Orders(ono)
```

```
);

create table Manager(
    mno int auto_increment primary key ,
    mname varchar(20) not null ,
    mpassword varchar(20) not null
);

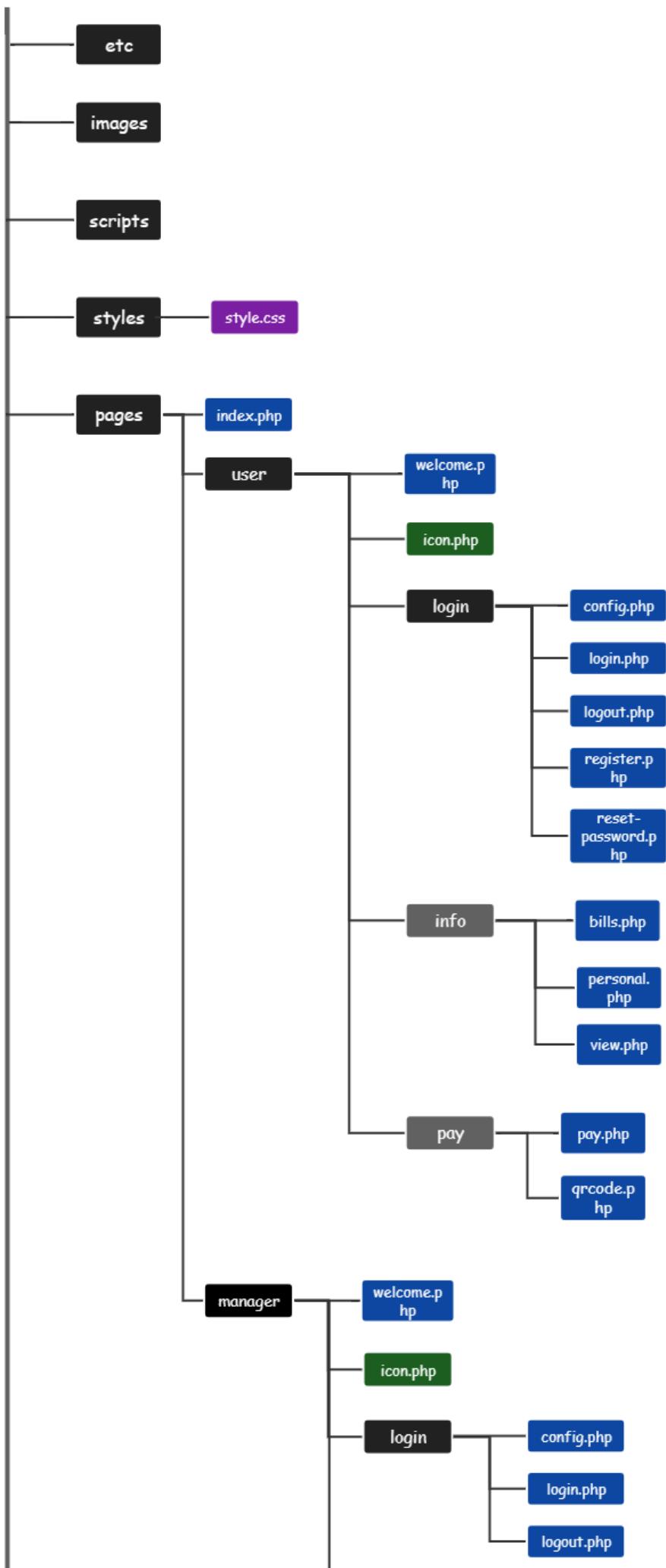
insert into Manager(mname, mpassword) values ('manager','manager');
```

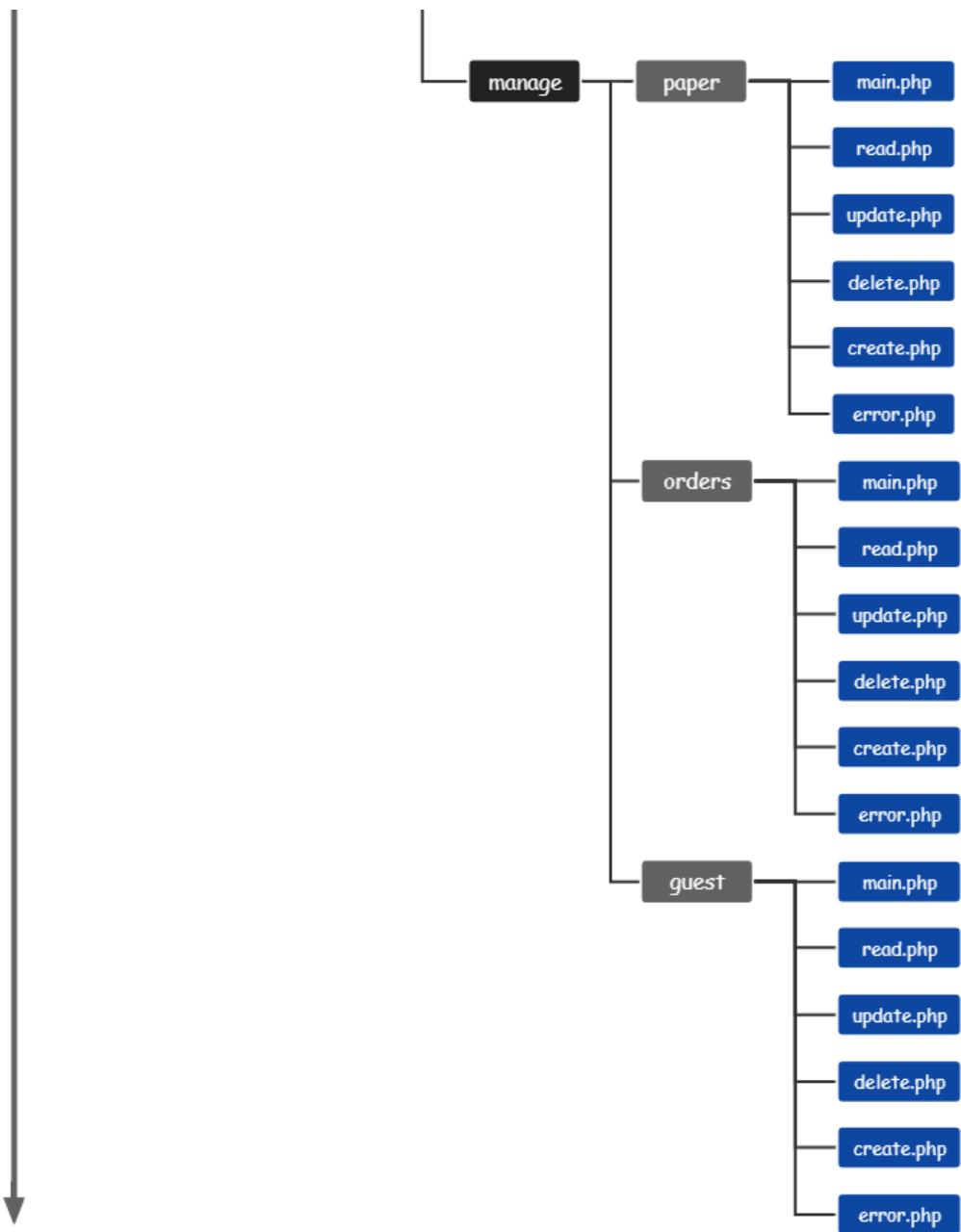
五、系统功能实现

5.1 系统功能模块

本系统的功能模块主要包括：用户登录、用户注册、用户信息修改、订阅报纸、查询订阅信息、查询订阅账单、查询报纸信息、管理员登录、管理员信息修改、添加报纸、删除报纸、修改报纸信息、查询报纸信息、查询订阅信息、查询订阅账单、退出系统。

整体网页文件的目录结构如下：





下面将对这些功能的实现展开具体介绍。

5.2 用户功能实现

5.2.0 数据库连接

在实现所有功能之前，首先需要连接原先创建好的数据库。在 `config.php` 中实现这一功能：

```

<?php
define('DB_SERVER', '127.0.0.1:40000');
define('DB_USERNAME', 'manager');
define('DB_PASSWORD', 'manager');
define('DB_NAME', 'SubscriptionSystem');

/* Attempt to connect to MySQL database */
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Check connection
if($link === false){

```

```
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

5.2.1 用户登录

用户登录模块的实现主要包括以下几个步骤：

1. 用户输入用户名和密码，点击登录按钮；
2. 系统将用户输入的用户名和密码与数据库中的数据进行比较，如果用户名和密码正确，则进入用户主界面，否则提示用户名或密码错误。

我们将在动态网页中创建一个表单，用户输入用户名和密码，点击登录按钮后，将表单中的数据传递给服务器，服务器再将数据与数据库中的数据进行比较。在实现中，我们使用 `$_SESSION` 存储用户信息、进行会话跟踪，以便在用户登录后，能够在整个会话中保持登录状态。

这部分功能在 `login.php` 文件中实现，具体代码如下：

```
<?php
// Initialize the session
session_start();

// Check if the user is already logged in, if yes then redirect him to welcome
page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
    header("location: ../welcome.php");
    exit;
}

// Include config file
require_once "config.php";

// Define variables and initialize with empty values
$name = $password = "";
$name_err = $password_err = $login_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Check if name is empty
    if(empty(trim($_POST["name"]))){
        $name_err = "Please enter name.";
    }
    else{
        $name = trim($_POST["name"]);
    }

    // Check if password is empty
    if(empty(trim($_POST["password"]))){
        $password_err = "Please enter your password.";
    } else{
        $password = trim($_POST["password"]);
    }

    // validate credentials
}
```

```

if(empty($name_err) && empty($password_err)){
    // Prepare a select statement
    $sql = "SELECT * FROM Manager WHERE mname = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_name);

        // Set parameters
        $param_name = $name;

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            // Store result
            mysqli_stmt_store_result($stmt);

            // Check if name exists, if yes then verify password
            if(mysqli_stmt_num_rows($stmt) == 1){
                // Bind result variables
                mysqli_stmt_bind_result($stmt, $mno, $mname, $mpassword);
                if(mysqli_stmt_fetch($stmt)){
                    if($password == $mpassword){
                        // Password is correct, so start a new session
                        session_start();

                        // Store data in session variables
                        $_SESSION["loggedin"] = true;
                        $_SESSION["id"] = $mno;
                        $_SESSION["name"] = $mname;
                        // Redirect user to welcome page
                        header("location: ../welcome.php");
                    } else{
                        // Password is not valid, display a generic error
                        $login_err = "Invalid name or password.";
                    }
                }
            } else{
                // name doesn't exist, display a generic error message
                $login_err = "Invalid name or password.";
            }
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }

        // Close statement
        mysqli_stmt_close($stmt);
    }
}

// Close connection
mysqli_close($link);
}
?>

```

5.2.2 用户注册

用户注册模块的实现主要包括以下几个步骤：

1. 用户输入用户名、密码、确认密码、邮箱地址、手机号码、地址和邮编，点击注册按钮；
2. 检查用户输入的数据是否合法，如果不合法，则提示用户重新输入；
3. 规定单个手机号只能注册一次，因此首先在数据库中查找具备相同手机号的用户，如果存在，则提示用户重新输入手机号；
4. 否则将用户输入的数据插入到数据库中，注册成功。

同样使用表单获取用户输入的数据，然后使用如下代码来实现上述功能：

```
<?php
// Initialize the session
session_start();

require_once "config.php";

// Check if the user is already logged in, if yes then redirect him to welcome
page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
    header("location: ../welcome.php");
    exit;
}

// Define variables and initialize with empty values
$telephone = $password = "";
$telephone_err = $password_err = $login_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Check if telephone is empty
    if(empty(trim($_POST["telephone"]))){
        $telephone_err = "Please enter telephone.";
    }
    elseif(!is_numeric($_POST['telephone'])){
        $telephone_err = "Please enter a valid telephone number.";
    }
    else{
        $telephone = trim($_POST["telephone"]);
    }

    // Check if password is empty
    if(empty(trim($_POST["password"]))){
        $password_err = "Please enter your password.";
    } else{
        $password = trim($_POST["password"]);
    }

    // validate credentials
    if(empty($telephone_err) && empty($password_err)){
        // Prepare a select statement
        $sql = "SELECT * FROM Guest WHERE gtelephone = ?";
```

```

if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt, "s", $param_telephone);

    // Set parameters
    $param_telephone = $telephone;

    // Attempt to execute the prepared statement
    if(mysqli_stmt_execute($stmt)){
        // Store result
        mysqli_stmt_store_result($stmt);

        // Check if telephone exists, if yes then verify password
        if(mysqli_stmt_num_rows($stmt) == 1){
            // Bind result variables
            mysqli_stmt_bind_result($stmt, $id, $name, $hashed_password,
$email, $address, $telephone, $postcode);
            if(mysqli_stmt_fetch($stmt)){
                if(password_verify($password, $hashed_password)){
                    // Password is correct, so start a new session
                    session_start();

                    // Store data in session variables
                    $_SESSION["loggedin"] = true;
                    $_SESSION["id"] = $id;
                    $_SESSION["name"] = $name;
                    $_SESSION["email"] = $email;
                    $_SESSION["address"] = $address;
                    $_SESSION["telephone"] = $telephone;

                    $_SESSION["postcode"] = $postcode;

                    // Redirect user to welcome page
                    header("location: ../welcome.php");
                } else{
                    // Password is not valid, display a generic error
                    message
                        $login_err = "Invalid telephone or password.";
                    }
                }
            } else{
                // telephone doesn't exist, display a generic error message
                $login_err = "Invalid telephone or password.";
            }
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }
        // Close statement
        mysqli_stmt_close($stmt);
    }
}
?>

```

5.2.3 查看报纸信息

用户登录后将会跳转到 `welcome.php`，该页面显示了用户可以进行的操作。用户可以点击“View and Subscribe Newspaper”按钮来查看报纸信息。点击按钮后将会跳转到 `view.php` 中，这个文件创建了一个表格，表格上列出了所有报纸的信息。从数据库中获取报纸信息的代码如下：

```
<?php
    // get the newspaper information from the database
$tableName = "Paper";
$columns = ["pno", "pname", "pprice", "frequency", "pwidth", "pheight",
"ppublisher"];
$PaperInfor = fetchData($tableName, $columns);
function fetchData($tableName, $columns){
    global $link;
    if(empty($link)){
        $msg = "Database connection failed";
        return $msg;
    }
    $columnName = implode(", ", $columns);
    $sql = "SELECT ".$columnName." FROM ".$tableName." ORDER BY pno";
    $result = mysqli_query($link, $sql);
    if($result == true) {
        if($result->num_rows > 0){
            $row = mysqli_fetch_all($result, MYSQLI_ASSOC);
            $msg = $row;
        }
        else{
            $msg = "No Newspaper";
        }
    }
    else{
        $msg = mysqli_error($link);
    }
    return $msg;
}
?>
```

5.2.4 订阅报纸

在上面提到的 `view.php` 中，表格的最后两列是 `Number` 和 `Period`，用户可以在这两列的输入框中输入订阅报纸的数量和订阅报纸的期数。之后点击 `submit` 按钮后，将会把用户的订阅信息插入到数据库中并跳转到结账页面。新增订阅信息的代码如下：

```
<?php
global $link;
global $Orders;
// get the added subscription from the page
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $gno = $_SESSION['id'];
    $onum = 0;
    $Orders = array(); // collect the orders in this bill

    // before adding orders, we need to get the correct one
```

```

$sql = "SELECT MAX(CAST(ono AS UNSIGNED)) FROM Orders";
if($stmt = mysqli_prepare($link, $sql)){
    mysqli_stmt_execute($stmt);
    mysqli_stmt_store_result($stmt);
    mysqli_stmt_bind_result($stmt, $onum);
    mysqli_stmt_fetch($stmt);
    mysqli_stmt_close($stmt);
}
// $onum = $onum + 1;

// before adding bills, we need to get the datetime
$now = "";
$sql = "SELECT NOW() AS NOW;";
if($stmt = mysqli_prepare($link, $sql)){
    mysqli_stmt_execute($stmt);
    mysqli_stmt_store_result($stmt);
    mysqli_stmt_bind_result($stmt, $now);
    mysqli_stmt_fetch($stmt);
    mysqli_stmt_close($stmt);
}
// now onum is the number of total orders
// we need to add 1 to get the ono of a new order

// add a flag to judge if add orders
$change = 0;

if(gettype($PaperInfor) != "string"){
    // if the paper information is not empty
    foreach($PaperInfor as $row){
        $pno = $row['pno'];
        $addNumber = $_POST["number".$pno];
        $addPeriod = $_POST["period".$pno];
        if($addNumber != "0"){
            $change = 1;
            // add an order to the database
            $sql = "INSERT INTO Orders (ono, gno, pno, onumber, period
VALUES (?, ?, ?, ?, ?, ?)";
            if($stmt = mysqli_prepare($link, $sql)){
                // bind variables
                mysqli_stmt_bind_param($stmt, "ssssi", $param_ono,
                $param_gno, $param_pno, $param_onumber, $param_period);
                // set parameters
                // echo "onum". $onum;
                $param_ono = $onum + 1;
                $param_gno = $gno;
                $param_pno = $row['pno'];
                $param_onumber = $addNumber;
                $param_period = $addPeriod;
                if(mysqli_stmt_execute($stmt)){
                    array_push($Orders, $param_ono);
                    $onum = $onum + 1;
                    echo "Order added successfully.";
                }
            } else{
                // print error message
            }
        }
    }
}

```

```

        echo mysqli_stmt_error($stmt);
        echo "Something went wrong when try to add order.";
    }

    mysqli_stmt_close($stmt);
}

$sql = "INSERT INTO bill (btime, ono) VALUES (?, ?)";
if($stmt = mysqli_prepare($link, $sql)){
    mysqli_stmt_bind_param($stmt, "ss", $para_datetime,
$param_ono);
    $para_datetime = $now;
    // echo $para_datetime;
    if(mysqli_stmt_execute($stmt)){
        echo "<br>Bill added successfully.<br>";
    }
    else{
        echo mysqli_stmt_error($stmt);
        echo "<br>Something went wrong when try to add
bill<br>";
    }
}
else{
    if($change == 1){
        // jump to the pay page
        foreach($Orders as $order){
            echo $order;
        }
        $_SESSION['Orders'] = $Orders;
        header("location: ../pay/pay.php");
    }
    else{
        echo "<span>do not add any subscription</span>";
    }
}
else{
    echo '<span class="invalid-feedback"><?php echo $email_err; ?>
</span>';
}
}
?>
```

值得一提的是，这里的 \$onum 是一个全局变量，用来记录当前的订单号，每次添加订单时，都会自动加 1。在首次使用 \$onum，通过 `SELECT MAX(CAST(ono AS UNSIGNED)) FROM Orders;` 的值加1来设置 `onum`，即使管理员在数据库中删除了订单，也能保证订单号的唯一性。

5.2.5 结账

上面在 `view.php` 中，用户点击 `Submit` 按钮后，会跳转到结账页面 `pay`。我们使用 `$_SESSION['Orders']` 存放用户的所有订单号，然后在结账页面中，通过遍历 `$_SESSION['Orders']` 中的订单号来生成账单。结账页面的代码如下：

```
<?php
global $link;
```

```

$Orders = $_SESSION['Orders'];
$total_price = 0;
foreach($Orders as $ono){
    //get the information of the order
    $sql = "SELECT pno, pname, pprice, frequency, pwidth, pheight,
ppublisher, onumber, period, pprice*frequency*onumber*period AS sum FROM Orders
NATURAL JOIN Paper WHERE ono = ?";
    if($stmt = mysqli_prepare($link, $sql)){
        mysqli_stmt_bind_param($stmt, "s", $para_ono);
        $para_ono = $ono;
        if(mysqli_stmt_execute($stmt)){
            mysqli_stmt_store_result($stmt);
            mysqli_stmt_bind_result($stmt, $pno, $pname, $pprice,
$frequency, $pwidth, $pheight, $ppublisher, $onumber, $period, $sum);
            if(mysqli_stmt_fetch($stmt)){
                $total_price += $sum;
                echo "<tr>";
                echo "<td>" . $ono . "</td>";
                echo "<td>" . $pname . "</td>";
                echo "<td>" . $pprice . "</td>";
                echo "<td>" . $frequency . "</td>";
                echo "<td>" . $pwidth . "</td>";
                echo "<td>" . $pheight . "</td>";
                echo "<td>" . $ppublisher . "</td>";
                echo "<td>" . $onumber . "</td>";
                echo "<td>" . $period . "</td>";
                echo "<td>" . $sum . "</td>";
                echo "</tr>";
            }
        }
    }
}

?>

```

(以上只截取了代码文件中最为关键的部分，完整代码将在后面给出代码仓库)

5.2.6 查看历史账单

在 `welcome.php` 中，用户点击 `View Your Bills` 按钮后，会跳转到 `bills.php` 页面，该页面展示了用户的所有历史账单。为了使页面更为规整，我们将账单按照时间进行了分组，即同一时间的账单会被放在同一个表格中。我们首先通过 `SELECT distinct btime FROM Bill NATURAL JOIN Orders WHERE gno = ?;` 语句获取到所有的账单时间，然后通过遍历账单时间，通过 `SELECT * FROM Orders NATURAL JOIN Paper NATURAL JOIN Bill WHERE gno = ? AND btime = ?` 语句获取到该时间的所有账单，最后将同一时间的账单放在同一个表格中。

该页面的代码如下：

```

<!-- get the bills' datetime -->
<?php
global $link;
$bdatetimes = "";
if(empty($link)){

```

```

        echo "Database connection failed";
        return;
    }
    $para_gno = $_SESSION["id"];
    $sql = "select distinct btime from Bill natural join Orders where gno =
    ".$para_gno.";";
    $result = mysqli_query($link, $sql);
    if($result == true){
        if($result->num_rows > 0){
            $bdatetimes = mysqli_fetch_all($result, MYSQLI_ASSOC);
        }
    }
    else{
        echo "Error: " . mysqli_error($link);
    }
?>

<!-- print all bills -->
<?php
if(gettype($bdatetimes) == "string"){
    echo "No bills";
    return;
}
else{
    $total_price = 0;
    foreach($bdatetimes as $btime){
        echo "<div class='wrapper'></div>";
        echo "<h2>Bill Time: ".$btime["btime"]."</h2>";
        // get the orders in this bill
        $para_btime = $btime["btime"];
        $sql = "select * from Bill natural join Orders natural join Paper
where gno = ".$para_gno." and btime = '".$para_btime."' order by ono;";
        $result = mysqli_query($link, $sql);
        if($result == true){
            if($result->num_rows > 0){
                $orders = mysqli_fetch_all($result, MYSQLI_ASSOC);
            }
        }
        else{
            echo "Error: " . mysqli_error($link);
        }
        // print the orders
        echo "<table class='table table-bordered table-striped'>";
        echo "<thead>";
        echo "<tr>";
        echo "<th>Order ID</th>";
        echo "<th>Newspaper Name</th>";
        echo "<th>Price</th>";
        echo "<th>Frequency</th>";
        echo "<th>Number</th>";
        echo "<th>Period</th>";
        echo "<th>Sum Price</th>";
        echo "</tr>";
        echo "</thead>";
        echo "<tbody>";

```

```

        foreach($orders as $order){
            $sumprice = $order['pprice'] * $order['frequency'] *
$order['onumber'] * $order['period'];
            echo "<tr>";
            echo "<td>" . $order['ono'] . "</td>";
            echo "<td>" . $order['pname'] . "</td>";
            echo "<td>" . $order['pprice'] . "</td>";
            echo "<td>" . $order['frequency'] . "</td>";
            echo "<td>" . $order['onumber'] . "</td>";
            echo "<td>" . $order['period'] . "</td>";
            echo "<td>" . $sumprice. "</td>";
            echo "</tr>";
            $total_price += $sumprice;
        }
        echo "</tbody>";
        echo "</table>";
        // print the total price of this bill
        echo "<p class='price'>Total Price: ".$total_price."</h2>";
    }
}

?>

```

5.2.7 查看个人信息

在 `welcome.php` 中点击 `View Your Information` 将会跳转到 `Personal.php` 页面，该页面展示了用户的个人信息。我们通过 `SELECT * FROM Guest WHERE gno = ?` 语句获取到用户的个人信息，然后将信息展示在页面中。

关键代码如下：

```

<?php
    global $link;
    $pno = $_SESSION['id'];
    $sql = "SELECT gname, gemail, gaddress, gtelephone, gpostcode FROM Guest
WHERE gno = ?";
    if($stmt = mysqli_prepare($link, $sql)){
        mysqli_stmt_bind_param($stmt, "s", $para_pno);
        $para_pno = $pno;
        if(mysqli_stmt_execute($stmt)){
            mysqli_stmt_store_result($stmt);
            mysqli_stmt_bind_result($stmt, $pname, $pemail, $paddress,
$ptelephone, $ppostcode);
            if(mysqli_stmt_fetch($stmt)){
                echo "<tr><th>name</th><td>".$pname."</td></tr>";
                echo "<tr><th>email</th><td>".$pemail."</td></tr>";
                echo "<tr><th>address</th><td>".$paddress."</td></tr>";
                echo "<tr><th>telephone</th><td>".$ptelephone."</td></tr>";
                echo "<tr><th>postcode</th><td>".$ppostcode."</td></tr>";
            }
        }else{
            echo "fail to fetch information";
        }
    }
}

```

```

        }
    else{
        echo "fail to execute the statement";
    }
}
else{
    echo "fail to prepare the statement";
}
?>

```

至此，我们完成了用户主要功能的实现。实际出于个人兴趣我还实现了重置密码和退出登录的功能，这些功能并不在本课程设计的要求范围内，因此这里不对具体实现细节进行展开。

5.3 管理员功能

5.3.1 登录和主页

由于此处的登录和用户登录的实现类似，不再重复说明。

管理员登录后，将会跳转到 `welcome.php` 页面，该页面展示了管理员的主要功能。管理员可以通过点击页面上的按钮来实现对应的功能。

这些功能包括：

- 用户信息管理；
- 订单信息管理；
- 报纸信息管理；

每个管理都将实现对应的增删改查功能，这里将以用户信息管理为例展开说明。

5.3.2 用户信息管理

点击 `Manage Guest Info` 将跳转到 `guest/main.php` 页面，该页面展示了所有用户的信息。每行对应一位用户，每行的最后一列包含了对该用户信息的查看、修改、删除操作。

在表格最底端，我们还提供了添加用户的功能，点击 `Add New Guest` 将跳转到 `create.php` 页面，该页面包含了添加用户的表单，用户填写完表单后点击 `submit` 按钮即可将用户信息添加到数据库中。

下面展示上述功能实现的部分代码：

1. 查看用户信息

```

<?php
if(isset($_GET["gno"]) && !empty(trim($_GET["gno"]))){
    require_once "../login/config.php";
    $sql = "SELECT * FROM Guest WHERE gno = ?";
    if($stmt = mysqli_prepare($link, $sql)){
        mysqli_stmt_bind_param($stmt, "s", $param_id);
        $param_id = trim($_GET["gno"]);
        if(mysqli_stmt_execute($stmt)){
            $result = mysqli_stmt_get_result($stmt);

            if(mysqli_num_rows($result) == 1){
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                $name = $row["gname"];
                $email = $row["gemail"];
            }
        }
    }
}

```

```

        $address = $row["gaddress"];
        $telephone = $row["gtelephone"];
        $postcode = $row["gpostcode"];
    } else{
        header("location: error.php");
        exit();
    }
} else{
    echo "Oops! Something went wrong. Please try again later.";
}
}

mysqli_stmt_close($stmt);
} else{
    header("location: error.php");
    exit();
}
?>

```

2. 修改用户信息

```

<?php
// Define variables and initialize with empty values
$name = $email = $address = $telephone = $postcode = "";
$name_err = $email_err = $address_err = $telephone_err = $postcode_err = "";

// Processing form data when form is submitted
if(isset($_POST["gno"]) && !empty($_POST["gno"])){
    // Get hidden input value
    $gno = $_POST["gno"];

    // ...
    // validate input

    if(empty($name_err) && empty($price_err) && empty($frequency_err)){
        $sql = "UPDATE Guest SET gname=?, gemail=?, gaddress=?, gtelephone=?,
gpostcode=? WHERE gno=?";
        if($stmt = mysqli_prepare($link, $sql)){
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "ssssss", $name, $email, $address,
$telephone, $postcode, $gno);
            // Attempt to execute the prepared statement
            if(mysqli_stmt_execute($stmt)){
                header("refresh:1; url = main.php");
                exit();
            } else{
                echo mysqli_stmt_error($stmt);
                echo "Oops! Something went wrong. Please try again later.";
            }
        }
        mysqli_stmt_close($stmt);
    }
}
?>

```

3. 删除用户信息

```

<?php
// Process delete operation after confirmation
if(isset($_POST["id"]) && !empty($_POST["id"])){
    $sql = "DELETE FROM Guest WHERE gno = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        mysqli_stmt_bind_param($stmt, "i", $param_id);

        $param_id = trim($_POST["id"]);

        if(mysqli_stmt_execute($stmt)){
            header("location: main.php");
            exit();
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }
    }

    mysqli_stmt_close($stmt);
}

} else{
    // Check existence of id parameter
    if(empty(trim($_GET["gno"]))){
        header("location: error.php");
        exit();
    }
}
?>

```

4. 添加用户信息

```

<?php
$id = $name = $price = $frequency = $width = $height = $publisher = "";
$name_err = $price_err = $frequency_err = $width_err = $height_err =
$publisher_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // ...
    // validate inputs

    // Check input errors before inserting in database
    if(empty($name_err) && empty($price_err) && empty($frequency_err)){
        // firstly generate a new paper id
        $id = 0;
        $sql = "SELECT MAX(pno) FROM Paper";
        $stmt = mysqli_prepare($link, $sql);
        if($stmt) {
            if(mysqli_stmt_execute($stmt)){
                mysqli_stmt_store_result($stmt);
                if(mysqli_stmt_num_rows($stmt) == 1){
                    mysqli_stmt_bind_result($stmt, $id);
                    if(mysqli_stmt_fetch($stmt)){

```

```

        $id = $id + 1;
        echo $id;
    }
}
}

mysqli_stmt_close($stmt);

// then add the newspaper info
$sql = "INSERT INTO Paper(pno, pname, pprice, frequency, pwidth,
pheight, ppublisher) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

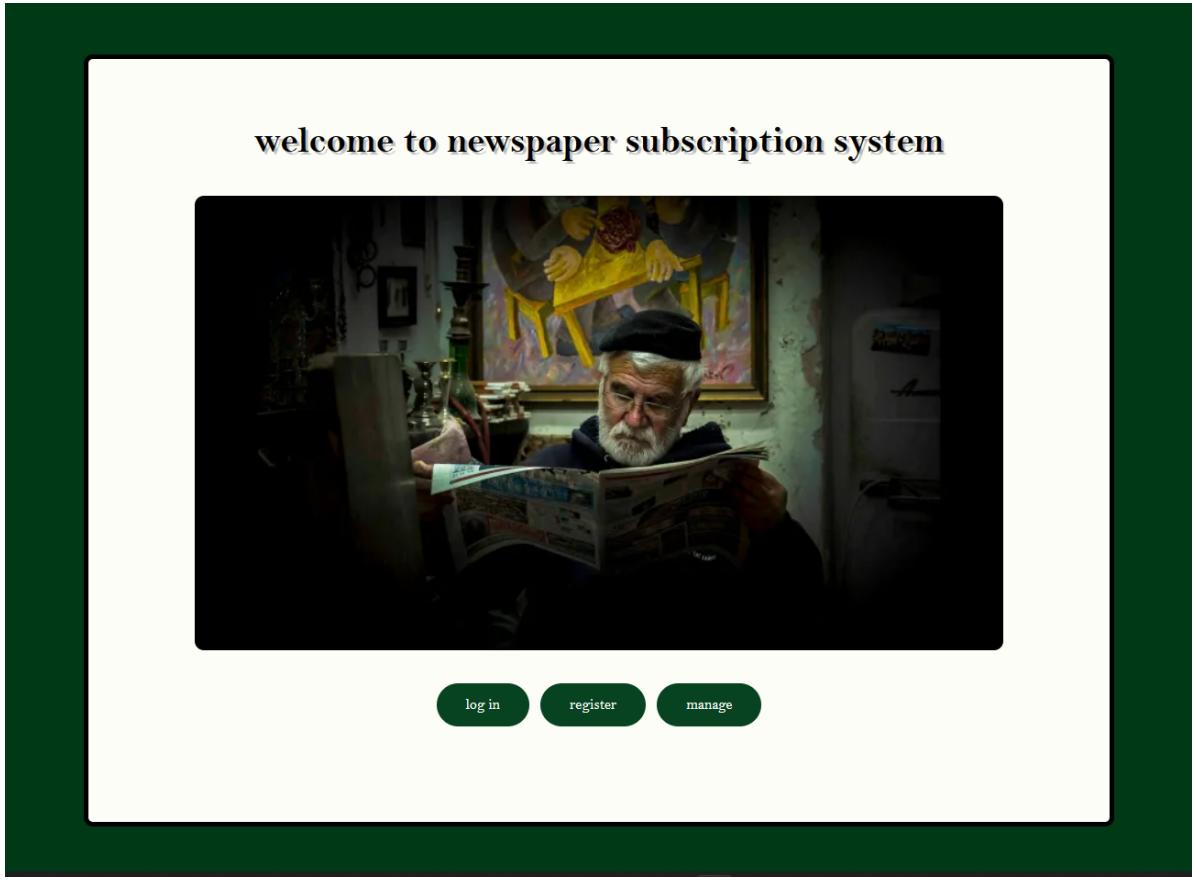
if($stmt = mysqli_prepare($link, $sql)){
    mysqli_stmt_bind_param($stmt, "sssssss", $id, $name, $price,
$frequency, $width, $height, $publisher);

    if(mysqli_stmt_execute($stmt)){
        header("location: main.php");
        exit();
    } else{
        echo "Oops! Something went wrong. Please try again later.";
    }
}
mysqli_stmt_close($stmt);
}

?>
```

六、实验结果

6.0 初始界面

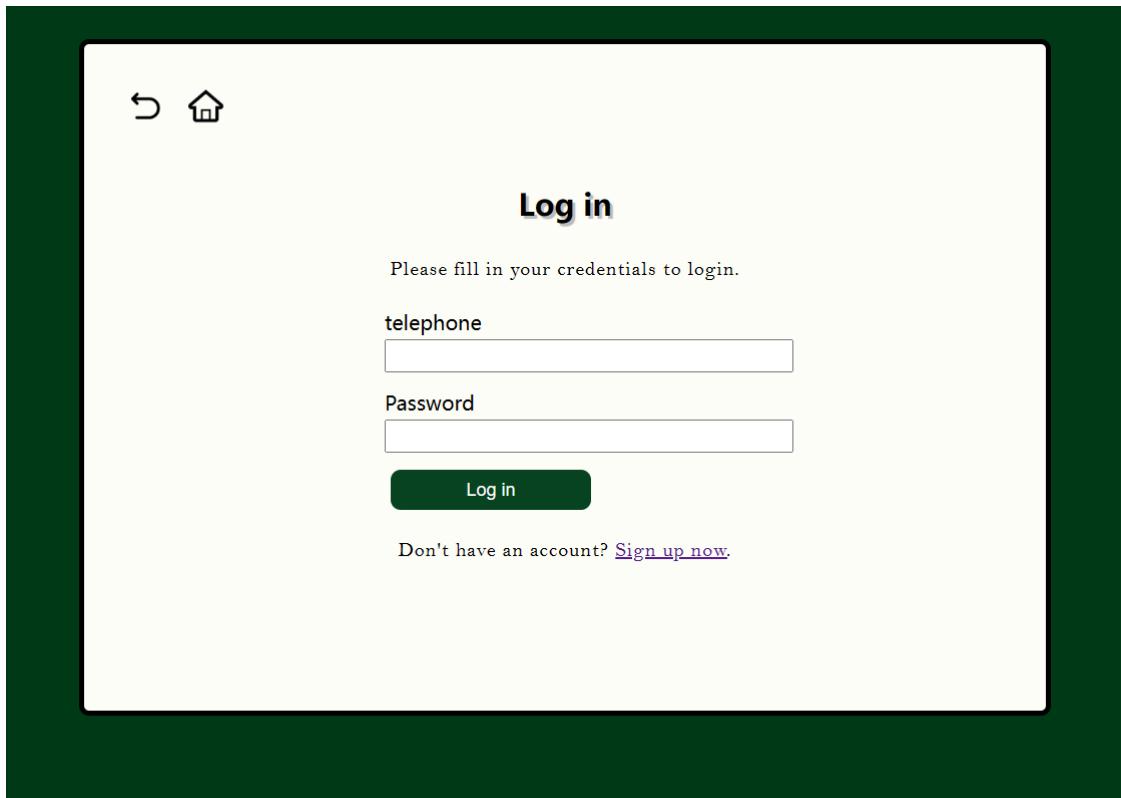


6.1 用戶界面

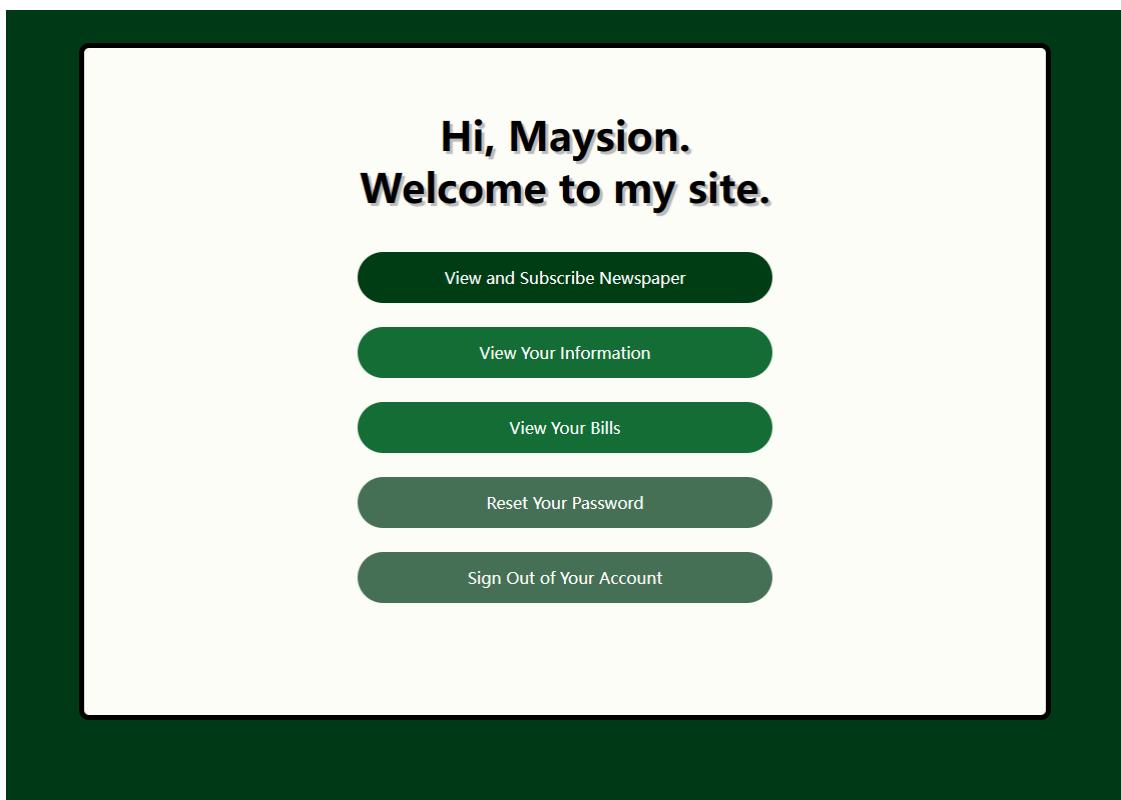
6.1.1 用戶註冊

The image displays a "Sign Up" form for creating a new account. The title "Sign Up" is centered at the top in a bold, black font. Below it, a small instruction reads "Please fill this form to create an account." The form consists of several input fields with labels: "telephone", "Username", "Password", "Confirm Password", "email", "address", and "postcode". Each label is followed by a text input field. At the bottom of the form are two buttons: "Submit" (in green) and "Reset" (in grey). Below these buttons, a link "Already have an account? [Login here.](#)" is provided.

6.1.2 用戶登錄



6.1.3 用户主页



6.1.4 查看报纸和添加订阅

↶ ↷

view and purchase

no	name	price	frequency	width	height	publisher	Number	Period(months)
1	New York Time	1.50	\$0	0	0	publisher1	0	1
2	paper2	1.50	\$0	500	450	publisher2	0	1
3	paper3	2.20	\$0			publisher3	0	1
4	paper4	3.50	\$0			publisher4	0	1
5	paper5	1.10	15				0	1

Submit

6.1.5 提交订单与支付

Check Your Bill

no	name	price	frequency	width	height	publisher	Number	Period(months)	Sum
83	New York Time	1.50	\$0	0	0	publisher1	4	1	180.00
84	paper2	1.50	\$0	500	450	publisher2	4	2	\$12.00
85	paper3	2.20	\$0			publisher3	1	1	66.00

Total Price: 558

Pay



6.1.6 查看历史订单

Your Bills						
Bill Time: 2022-12-17 16:14:08						
Order ID	Newspaper Name	Price	Frequency	Number	Period	Sum Price
54	New York Time	1.50	30	5	1	15.0
55	paper2	1.50	30	9	1	13.5
56	paper3	2.20	30	5	4	11.0
57	paper4	3.80	30	1	2	1.90
58	paper5	1.10	15	1	1	1.10

Total Price: 20.20

Bill Time: 2022-12-18 03:29:22						
Order ID	Newspaper Name	Price	Frequency	Number	Period	Sum Price
59	New York Time	1.50	30	5	1	15.0
60	paper2	1.50	30	9	1	13.5
61	paper3	2.20	30	5	4	11.0
62	paper4	3.80	30	1	2	1.90
63	paper5	1.10	15	1	1	1.10

Total Price: 40.40

6.1.7 查看个人信息

Your Information

name	Maysion
email	test@mail.com
address	test address
telephone	123456789
postcode	123456

[Back](#)

6.1.8 修改密码

Reset Password

Please fill in this form to reset your password.

New Password

Confirm Password

[Submit](#) [Cancel](#)

6.1.9 退出登录

Good Bye!

6.2 管理员界面

6.2.1 管理员登录

The screenshot shows a login interface with a white background and a dark border. At the top center, the word "Log in" is displayed in a bold, black font. Below it, a smaller message reads "Please fill in your credentials to login." There are two input fields: one for "Name" containing the text "manager" and another for "Password" containing several dots. A green "Log in" button is positioned below the password field. At the bottom, there is a link in blue text that says "Not a manager? [Jump back and login as normal user.](#)"

6.2.2 管理员主页

**Hi, manager.
Let's do some management.**

Manage Newspaper

Manage Guest Info

Manage Orders

Log out

6.2.3 管理报纸



Newspaper Details

#	Name	Price	Frequency	Width	Width	Publisher	operation
1	New York Time	1.50	30	0	0	publisher1	
2	paper2	1.80	30	300	450	publisher2	
3	paper3	2.20	30			publisher3	
4	paper4	3.50	30			publisher4	
5	paper5	1.10	15				

Add New Newspaper

新增报纸:

Add Newspaper Info

Please fill this form and submit to add Newspaper record to the database.

name

price

frequency

width

height

publisher

SubmitCancel

修改报纸信息：

Update Record

Please edit the input values and submit to update the Newspaper record.

Name	<input type="text" value="update name"/>
price	<input type="text" value="1.50"/>
frequency	<input type="text" value="25"/>
width	<input type="text" value="22"/>
height	<input type="text" value="44"/>
publisher	<input type="text" value="一个出版社"/>

SubmitCancel

查看报纸信息：

View Paper Info

Name	update name
Price	1.50
Frequency	25
Width	22
Height	44
Publisher	一个出版社

Back

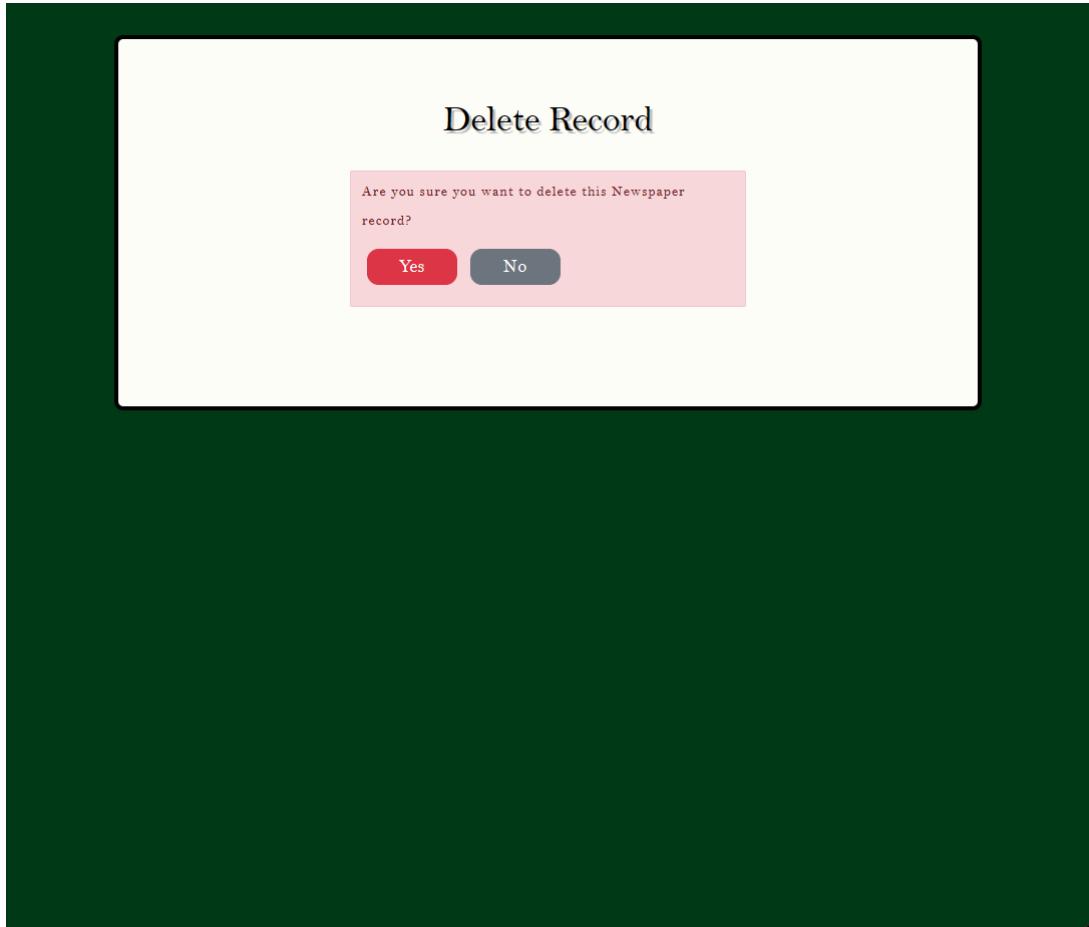
删除报纸：

↶ ↷

Newspaper Details

#	Name	Price	Frequency	Width	Width	Publisher	operation
1	New York Time	1.50	50	0	0	publisher1	
2	paper2	1.80	50	300	450	publisher2	
3	paper3	2.20	50			publisher3	
4	paper4	3.80	50			publisher4	
5	paper5	1.10	15				
6	update name	1.50	25	22	44	一个出版社	Delete Record

+ Add New Newspaper



6.2.4 管理用户

A screenshot of a guest information management interface. The title is "Guest Information". A table lists 9 guests with columns for #, Name, Email, Address, Telephone, Postcode, and Operation (with icons for edit, delete, and view). A purple button at the bottom left says "+ Add New Guest".

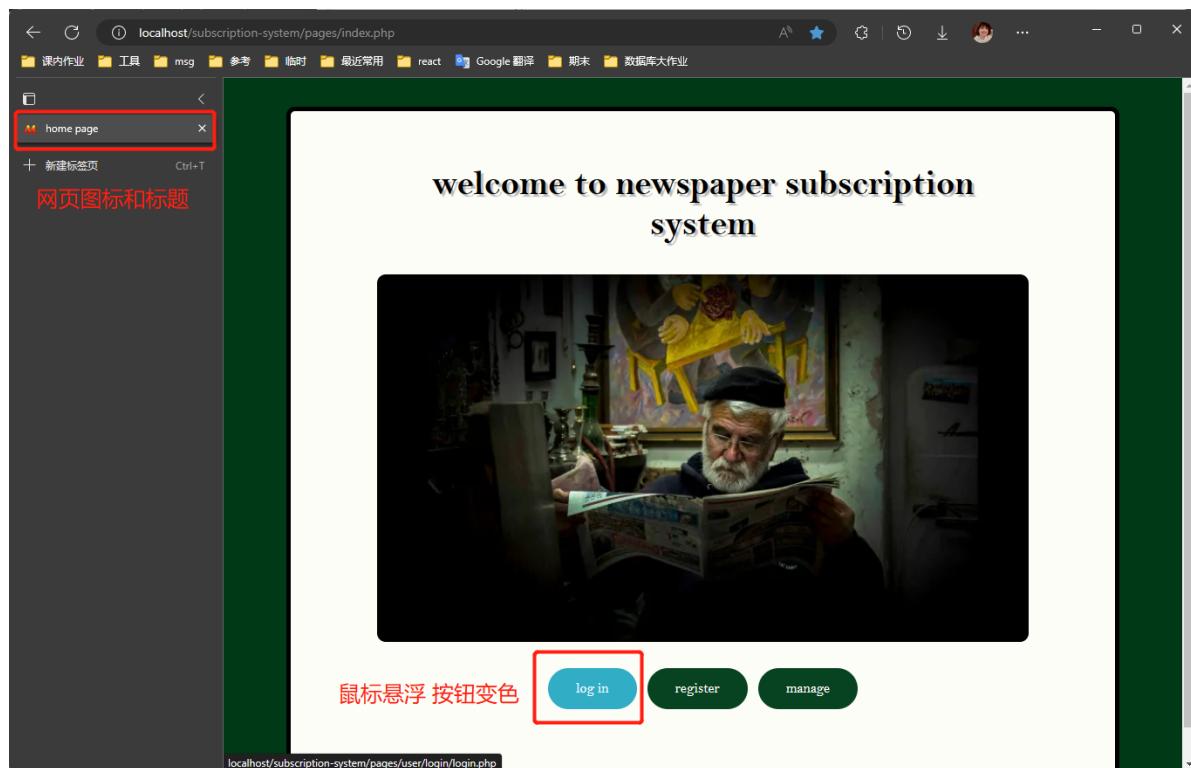
#	Name	Email	Address	Telephone	Postcode	Operation
1	小张	xiaozhang@mail.com	地球村	16677778888	666666	
2	小美	xiaomei@mail.com	北极村	123456123456	111111	
3	test	test@mail.com	test address	123456	123456	
4	test2	test@mail.com	test address	1234567	123456	
5	xiaoming	255@mail.com	xiaoming's home	255255255	123123	
6	xiaoxuan	xiaoxuan@mail.com	xiaoxuan's home	123123	123456	
7	maysion	test@mail.com	test address	12345678	123456	
8	Maysion	test@mail.com	test address	123456789	123456	
9	zhangmx	test@mail.com	test address	1234567890	123456	

+ Add New Guest

6.2.5 管理订单

#	Guest ID	Paper ID	Number	Period	Order Time	operation
10	S	4	1	1	2025-01-02 13:59:03	🔍✍️🗑️
10	S	4	1	1	2025-01-02 13:59:21	🔍✍️🗑️
10	S	4	1	1	2025-01-02 14:01:53	🔍✍️🗑️
10	S	4	1	1	2025-01-02 14:10:50	🔍✍️🗑️
10	S	4	1	1	2025-01-02 14:10:53	🔍✍️🗑️
11	S	1	1	1	2025-01-02 14:02:44	🔍✍️🗑️
11	S	1	1	1	2025-01-02 14:05:09	🔍✍️🗑️
11	S	1	1	1	2025-01-02 14:08:08	🔍✍️🗑️
11	S	1	1	1	2025-01-02 14:08:32	🔍✍️🗑️
14	S	1	1	1	2022-12-17 12:49:48	🔍✍️🗑️
15	S	1	2	1	2022-12-17 12:50:01	🔍✍️🗑️
16	S	1	2	1	2022-12-17 12:50:12	🔍✍️🗑️
17	S	1	1	1	2022-12-17 12:50:19	🔍✍️🗑️
18	S	2	1	1	2022-12-17 12:50:19	🔍✍️🗑️
19	S	3	1	1	2022-12-17 12:50:19	🔍✍️🗑️
20	S	1	1	1	2022-12-17 12:52:29	🔍✍️🗑️
21	S	2	1	1	2022-12-17 12:52:29	🔍✍️🗑️

其他细节展示：



左上角按钮：返回、跳转主页

Log in

Please fill in your credentials to login.

telephone
123456789

Password
123456789

Log in

点击显示或隐藏密码

Don't have an account? [Sign up now.](#)

Sign Up

Please fill this form to create an account.

telephone
This telephone is already taken. 注册时检测是否手机号已注册

Username
mayson

Password
Password must have at least 6 characters. 规定使用六位及以上密码

Confirm Password
...

email
...

address
Please enter a address. 必需项为空时出现警告

postcode
Please enter a postcode.

Submit **Reset**

Already have an account? [Login here.](#)

↶ ↷

view and purchase

no	name	price	frequency	width	height	publisher	Number	Period(months)
1	New York Time	1.50	\$0	0	0	publisher1	<input type="text" value="4"/>	<input type="text" value="1"/>
2	paper2	1.50	\$0	500	430	publisher2	<input type="text" value="4"/>	<input type="text" value="2"/>
3	paper3	2.20	\$0			publisher3	<input type="text" value="1"/>	<input type="button" value="0"/> 使用按钮或输入数字设定参数
4	paper4	3.50	\$0			publisher4	<input type="text" value="0"/>	<input type="text" value="1"/>
5	paper5	1.10	\$15				<input type="text" value="0"/>	<input type="text" value="1"/>

六、总结

在本次课程设计中，通过实现一个简单的报纸订阅系统，我深入学习和思考了数据库应用的六个步骤。在完成系统需求分析后，我明确了数据库中需要使用的数据对象和类型；通过绘制ER图，并将其转换为关系模式，我设计出了数据库的基本模型。考虑到数据库的实体设计，我最终建立了数据库的关系模型，并将其应用到网页的搭建和使用中。伴随着实践中出现的问题，我不断地进行调整和优化，在实现网页应用的同时，也不断深化了自己对数据库设计的理解。

七、说明

本系统的源代码已上传至GitHub，代码仓库为：

<https://github.com/Doriyz/subscription-system.git>

