Avvia xampp

Immagine che contiene testo, schermata, software, Icona del computer

Il contenuto generato dall'IA potrebbe non essere corretto.

1. Avviare Apache 🡪 da browser accedi qui: <http://localhost:89/dashboard/>
2. Avviare MySQL 🡪 da dbeaver usare la connessione phpMyAdmin

**1. Creazione e Popolamento del Database**

Esegui questo script SQL (su MySQL ad esempio) per creare il database “tennis” e le tabelle con alcuni dati di esempio:

USE tennis;

-- Creazione tabella utenti con ruolo e data di registrazione

CREATE TABLE utenti (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

cognome VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

ruolo ENUM('user', 'admin') DEFAULT 'user',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabella giocatori

CREATE TABLE giocatori (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

cognome VARCHAR(50) NOT NULL,

nazione VARCHAR(50) NOT NULL,

eta INT NOT NULL,

altezza INT NOT NULL, -- Altezza in centimetri

peso INT NOT NULL, -- Peso in chilogrammi

mano\_dominante ENUM('destra', 'sinistra') NOT NULL, -- Mano dominante

velocita\_servizio INT NOT NULL, -- Velocità massima del servizio (km/h)

potenza\_dritto INT NOT NULL, -- Rating (1-100) della potenza del dritto

precisione\_dritto INT NOT NULL, -- Rating (1-100) della precisione del dritto

potenza\_rovescio INT NOT NULL, -- Rating (1-100) della potenza del rovescio

precisione\_rovescio INT NOT NULL, -- Rating (1-100) della precisione del rovescio

mobilita INT NOT NULL, -- Rating (1-100) della mobilità in campo

resistenza INT NOT NULL, -- Rating (1-100) della resistenza fisica

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE giocatori

ADD COLUMN img VARCHAR(255) NOT NULL DEFAULT '';

-- Tabella ranking (classifiche)

CREATE TABLE ranking (

id INT NOT NULL AUTO\_INCREMENT,

giocatore\_id INT NOT NULL,

posizione INT NOT NULL,

punti INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (giocatore\_id) REFERENCES giocatori(id)

ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabella categoria con chiave primaria "id"

CREATE TABLE categoria (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabella tornei con riferimento alla categoria tramite "categoria\_id"

CREATE TABLE tornei (

id INT NOT NULL AUTO\_INCREMENT,

categoria\_id INT NOT NULL,

nome VARCHAR(100) NOT NULL,

luogo VARCHAR(100) NOT NULL,

data\_inizio DATE NOT NULL,

data\_fine DATE NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (categoria\_id) REFERENCES categoria(id)

ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabella partite

CREATE TABLE partite (

id INT NOT NULL AUTO\_INCREMENT,

torneo\_id INT NOT NULL,

giocatore1\_id INT NOT NULL,

giocatore2\_id INT NOT NULL,

risultato VARCHAR(20) DEFAULT NULL,

PRIMARY KEY (id),

FOREIGN KEY (torneo\_id) REFERENCES tornei(id)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (giocatore1\_id) REFERENCES giocatori(id)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (giocatore2\_id) REFERENCES giocatori(id)

ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabella biglietti

CREATE TABLE biglietti (

id INT NOT NULL AUTO\_INCREMENT,

torneo\_id INT NOT NULL,

prezzo DECIMAL(10,2) NOT NULL,

disponibilita INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (torneo\_id) REFERENCES tornei(id)

ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE biglietti\_utenti (

id INT NOT NULL AUTO\_INCREMENT,

utente\_id INT NOT NULL,

biglietto\_id INT NOT NULL,

quantita INT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

FOREIGN KEY (utente\_id) REFERENCES utenti(id)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (biglietto\_id) REFERENCES biglietti(id)

ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE news (

id INT NOT NULL AUTO\_INCREMENT,

titolo VARCHAR(255) NOT NULL,

descrizione TEXT NOT NULL,

img VARCHAR(255) NOT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Inserisci i giocatori

INSERT INTO giocatori (nome, cognome, nazione, eta, altezza, peso, mano\_dominante, velocita\_servizio, potenza\_dritto, precisione\_dritto, potenza\_rovescio, precisione\_rovescio, mobilita, resistenza) VALUES

('Jannik', 'Sinner', 'Italia', 22, 188, 75, 'destra', 210, 90, 88, 85, 87, 90, 88),

('Carlos', 'Alcaraz', 'Spagna', 21, 185, 70, 'destra', 205, 88, 86, 87, 85, 92, 90),

('Novak', 'Djokovic', 'Serbia', 36, 188, 77, 'destra', 200, 85, 90, 80, 90, 95, 95),

('Rafael', 'Nadal', 'Spagna', 37, 185, 85, 'sinistra', 195, 80, 85, 78, 80, 90, 98),

('Daniil', 'Medvedev', 'Russia', 28, 198, 90, 'destra', 215, 82, 88, 84, 86, 88, 85),

('Alexander', 'Zverev', 'Germania', 26, 198, 85, 'destra', 220, 87, 90, 85, 88, 86, 87),

('Stefanos', 'Tsitsipas', 'Grecia', 25, 185, 77, 'sinistra', 205, 84, 86, 83, 85, 89, 88),

('Matteo', 'Berrettini', 'Italia', 27, 190, 85, 'destra', 230, 92, 88, 80, 84, 84, 82),

('Casper', 'Ruud', 'Norvegia', 24, 185, 80, 'destra', 205, 83, 85, 82, 84, 87, 85),

('Felix', 'Auger-Aliassime', 'Canada', 23, 190, 80, 'destra', 210, 86, 84, 83, 82, 88, 87),

('Taylor', 'Fritz', 'USA', 26, 193, 88, 'destra', 230, 85, 83, 84, 82, 87, 86),

('Holger', 'Rune', 'Danimarca', 20, 180, 75, 'destra', 205, 82, 80, 81, 80, 90, 88);

UPDATE giocatori

SET img = 'https://images2.gazzettaobjects.it/assets-mc/tennis/giocatori/high/jannik\_sinner.png'

WHERE nome = 'Jannik' AND cognome = 'Sinner';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Carlos' AND cognome = 'Alcaraz';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Novak' AND cognome = 'Djokovic';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Rafael' AND cognome = 'Nadal';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Daniil' AND cognome = 'Medvedev';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Alexander' AND cognome = 'Zverev';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Stefanos' AND cognome = 'Tsitsipas';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Matteo' AND cognome = 'Berrettini';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Casper' AND cognome = 'Ruud';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Felix' AND cognome = 'Auger-Aliassime';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Taylor' AND cognome = 'Fritz';

UPDATE giocatori

SET img = 'images/giocatori/player.png'

WHERE nome = 'Holger' AND cognome = 'Rune';

-- Inserisci le classifiche

INSERT INTO ranking (giocatore\_id, posizione, punti) VALUES

(1, 4, 6780),

(2, 2, 7820),

(3, 1, 10000),

(4, 5, 5000),

(5, 3, 7500),

(6, 6, 4500),

(7, 7, 4000),

(8, 8, 3500),

(9, 9, 3000),

(10, 10, 2500),

(11, 11, 2000),

(12, 12, 1500);

-- Inserisci le categorie

INSERT INTO categoria (nome) VALUES

('Grand Slam'),

('ATP 500'),

('ATP 250');

-- Inserisci i tornei (ad es., associando entrambi alla categoria 'Grand Slam' con id = 1)

INSERT INTO tornei (categoria\_id, nome, luogo, data\_inizio, data\_fine) VALUES

(1, 'Wimbledon', 'Londra', '2025-06-28', '2025-07-11'),

(1, 'Roland-Garros', 'Parigi', '2024-05-26', '2024-06-09');

-- Inserimento di altri tornei che si svolgono nel 2026

INSERT INTO tornei (categoria\_id, nome, luogo, data\_inizio, data\_fine) VALUES

(1, 'US Open', 'New York', '2026-08-30', '2026-09-12'),

(1, 'Australian Open', 'Melbourne', '2026-01-15', '2026-01-28'),

(2, 'Internazionali BNL d''Italia', 'Roma', '2026-05-15', '2026-05-28');

-- Inserisci le partite

INSERT INTO partite (torneo\_id, giocatore1\_id, giocatore2\_id, risultato) VALUES

(1, 1, 2, '6:4;6:4'),

(2, 2, 1, '7:5;6:3');

-- Inserisci i biglietti

INSERT INTO biglietti (torneo\_id, prezzo, disponibilita) VALUES

(1, 120.50, 5000),

(2, 95.00, 3000);

-- Inserimento biglietti per i nuovi tornei (assumendo che i nuovi tornei abbiano ID 3, 4 e 5)

INSERT INTO biglietti (torneo\_id, prezzo, disponibilita) VALUES

(3, 110.00, 4000),

(4, 130.00, 3500),

(5, 100.00, 4500);

-- Inserisci gli utenti (le password sono già hashate con BCRYPT)

INSERT INTO utenti (nome, cognome, email, password, ruolo) VALUES

('Marco', 'Rossi', 'marco.rossi@email.com',

'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'user'),

('Luca', 'Bianchi', 'luca.bianchi@email.com',

'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'admin');

-- Inserimenti

INSERT INTO news (titolo, descrizione, img) VALUES

(

'Grande vittoria a Wimbledon',

'Durante un torneo estremamente competitivo, Jannik Sinner ha regalato agli appassionati di tennis una prestazione straordinaria, segnando una vittoria memorabile a Wimbledon. La partita, disputata con grande intensità, ha visto Sinner dominare il campo con precisione e grinta, superando avversari di alto livello. La sua velocità nei movimenti e la potenza del servizio hanno messo in difficoltà i migliori contendenti, evidenziando un mix di tecnica e determinazione che ha lasciato il pubblico entusiasta. Oltre a dimostrare una solida preparazione fisica e mentale, Sinner ha saputo adattarsi alle condizioni variabili dell’erba, confermando il suo status di stella emergente. Questa vittoria segna un punto di svolta nella sua carriera, aprendo la strada a ulteriori successi futuri.',

'images/news/wimbledon.jpg'

);

INSERT INTO news (titolo, descrizione, img) VALUES

(

'Emma Raducanu: non ne sapevo nulla',

'Nel corso di un''intervista esclusiva, Emma Raducanu ha rivelato dettagli inediti riguardanti la controversa causa PTPA contro ATP e WTA. La giovane tennista ha descritto come la vicenda abbia evidenziato dinamiche interne e problematiche mai discusse pubblicamente, sollevando interrogativi sul futuro del tennis professionistico. Con toni misurati, ha spiegato come il caso rappresenti un punto di svolta per il settore, sottolineando l''importanza di una maggiore trasparenza e riforma delle normative. La sua testimonianza ha fornito spunti interessanti, suscitando dibattiti tra gli addetti ai lavori e gli appassionati, e aprendo la strada a possibili riforme che potrebbero rivoluzionare il mondo del tennis.',

'https://www.tennisworlditalia.com/imgb/87679/-non-ne-sapevo-nulla--emma-raducanu-parla-della-causa-ptpa-contro-atp-e-wta.webp'

);

INSERT INTO news (titolo, descrizione, img) VALUES

(

'Medvedev domina sul campo',

'Durante un torneo di alto livello, Daniil Medvedev ha mostrato una prestazione eccezionale, dominando il campo con un mix di strategia, forza e precisione. In un match avvincente, il campione russo ha sfruttato ogni opportunità, imponendo il suo ritmo e gestendo abilmente ogni situazione critica. La sua performance, caratterizzata da colpi potenti e una notevole costanza, ha confermato la sua reputazione di protagonista assoluto del circuito. Medvedev ha lasciato il segno nel torneo, stabilendo nuovi standard di eccellenza e sottolineando la dedizione e disciplina che lo contraddistinguono. La sua vittoria ha entusiasmato gli appassionati, consolidando il suo status di leader e ispirando future generazioni nel mondo del tennis.',

'https://www.hindustantimes.com/ht-img/img/2025/03/21/550x309/Former-India-players-Somdev-Devvarman--pictured--a\_1742580031132.jpg'

);

INSERT INTO news (titolo, descrizione, img) VALUES

(

'Djokovic, l\'esperienza che conta',

'Novak Djokovic, uno dei veterani più titolati del tennis mondiale, ha dimostrato ancora una volta la sua straordinaria classe e determinazione in campo. Durante l''ultimo torneo, il campione serbo ha saputo gestire ogni fase della partita, combinando esperienza e talento per superare avversari di alto livello. La sua performance è stata un mix di tecnica raffinata e strategia ben calibrata, che gli ha permesso di controllare il ritmo del gioco in ogni momento decisivo. Djokovic ha confermato il suo status di leader nel panorama tennistico, evidenziando come l''esperienza maturata in anni di competizioni internazionali possa fare la differenza. La sua vittoria ha ispirato numerosi appassionati, rafforzando la sua leggenda nel mondo del tennis.',

'https://dims.apnews.com/dims4/default/bcc5a5f/2147483647/strip/true/crop/3047x1870+0+0/resize/599x368!/quality/90/?url=https%3A%2F%2Fassets.apnews.com%2Fb2%2F54%2Ffe2056f06e8f4543ef154a75c7d4%2Ff9250ac7d31f40ec81ed01fe5cd9ba82'

);

**Nota:** Per gestire i campi “foto” dei giocatori o altri attributi aggiuntivi (come le statistiche per il confronto head2head) potrai in seguito aggiornare la struttura della tabella giocatori oppure creare tabelle dedicate.

**2. Setup del Progetto CodeIgniter**

1. **Scarica e installa CodeIgniter:**  
   Scarica l’ultima versione di CodeIgniter dal sito ufficiale e scompatta il pacchetto nella directory del tuo server (es. htdocs/tennis\_project).
2. **Configurazioni iniziali:**
   * Imposta il parametro base\_url in application/config/config.php (es. http://localhost/tennis\_project/).
   * Abilita le sessioni e, se necessario, le funzioni di hashing per la gestione delle password.
3. **Configurazione del database:**  
   Modifica il file application/config/database.php per impostare la connessione al database “tennis”. Un esempio:

php

Copia

$db['default'] = array(

'dsn' => '',

'hostname' => 'localhost',

'username' => 'tuo\_username',

'password' => 'tua\_password',

'database' => 'tennis',

'dbdriver' => 'mysqli',

'dbprefix' => '',

'pconnect' => FALSE,

'db\_debug' => (ENVIRONMENT !== 'production'),

'cache\_on' => FALSE,

'cachedir' => '',

'char\_set' => 'utf8',

'dbcollat' => 'utf8\_general\_ci',

'swap\_pre' => '',

'encrypt' => FALSE,

'compress' => FALSE,

'stricton' => FALSE,

'failover' => array(),

'save\_queries' => TRUE

);

**3. Creazione dei Modelli (Models)**

**a. Modello per gli utenti**

**File:** application/models/Utenti\_model.php

php

Copia

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Utenti\_model extends CI\_Model {

public function \_\_construct(){

parent::\_\_construct();

$this->load->database();

}

// Recupera l'utente tramite email

public function get\_user\_by\_email($email){

$query = $this->db->get\_where('utenti', array('email' => $email));

return $query->row();

}

}

**b. Modello per le partite**

**File:** application/models/Partite\_model.php

php

Copia

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Partite\_model extends CI\_Model {

public function \_\_construct(){

parent::\_\_construct();

$this->load->database();

}

// Recupera le partite in cui la data\_inizio del torneo è nel futuro

public function get\_future\_matches(){

$today = date('Y-m-d');

$this->db->select('partite.\*, tornei.data\_inizio, tornei.nome as torneo\_nome');

$this->db->from('partite');

$this->db->join('tornei', 'tornei.id = partite.torneo\_id');

$this->db->where('tornei.data\_inizio >', $today);

$query = $this->db->get();

return $query->result();

}

}

**c. Modello per i giocatori**

**File:** application/models/Giocatori\_model.php

php

Copia

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Giocatori\_model extends CI\_Model {

public function \_\_construct(){

parent::\_\_construct();

$this->load->database();

}

// Cerca giocatori per nome o cognome

public function search($keyword = ''){

if($keyword){

$this->db->like('nome', $keyword);

$this->db->or\_like('cognome', $keyword);

}

$query = $this->db->get('giocatori');

return $query->result();

}

// Recupera il dettaglio di un giocatore tramite ID

public function get\_by\_id($id){

$query = $this->db->get\_where('giocatori', array('id' => $id));

return $query->row();

}

}

**Suggerimento:** Analogamente potrai creare modelli per le altre funzionalità (classifiche, head2head, biglietti) seguendo la medesima logica.

**4. Creazione dei Controller**

**a. Controller per il Login**

**File:** application/controllers/Login.php

php

Copia

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI\_Controller {

public function \_\_construct(){

parent::\_\_construct();

$this->load->model('Utenti\_model');

$this->load->library('session');

}

// Visualizza il form di login

public function index(){

$this->load->view('login\_view');

}

// Processa il form di login

public function process(){

$email = $this->input->post('email');

$password = $this->input->post('password');

$user = $this->Utenti\_model->get\_user\_by\_email($email);

if($user && password\_verify($password, $user->password)){

// Imposta i dati in sessione

$this->session->set\_userdata('user\_id', $user->id);

$this->session->set\_userdata('user\_role', $user->ruolo);

redirect('home');

} else {

$this->session->set\_flashdata('error', 'Credenziali non valide');

redirect('login');

}

}

// Logout

public function logout(){

$this->session->sess\_destroy();

redirect('login');

}

}

**b. Controller per la Home e le altre pagine**

**File:** application/controllers/Home.php

php

Copia

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI\_Controller {

public function \_\_construct(){

parent::\_\_construct();

// Se l'utente non è loggato, reindirizza al login

if(!$this->session->userdata('user\_id')){

redirect('login');

}

}

// Home page con navbar e messaggio di benvenuto

public function index(){

$data['title'] = "Home";

$data['welcome\_message'] = "Benvenuto nel sito di Tennis!";

$this->load->view('templates/header', $data);

$this->load->view('home\_view', $data);

$this->load->view('templates/footer');

}

// Pagina Partite: mostra le partite future

public function partite(){

$this->load->model('Partite\_model');

$data['partite'] = $this->Partite\_model->get\_future\_matches();

$data['title'] = "Partite Future";

$this->load->view('templates/header', $data);

$this->load->view('partite\_view', $data);

$this->load->view('templates/footer');

}

// Pagina Giocatori: ricerca e dettaglio giocatori

public function giocatori(){

$this->load->model('Giocatori\_model');

$search = $this->input->get('search');

$data['giocatori'] = $this->Giocatori\_model->search($search);

$data['title'] = "Ricerca Giocatori";

$this->load->view('templates/header', $data);

$this->load->view('giocatori\_view', $data);

$this->load->view('templates/footer');

}

// Metodo per visualizzare il dettaglio di un singolo giocatore

public function giocatore($id){

$this->load->model('Giocatori\_model');

$data['giocatore'] = $this->Giocatori\_model->get\_by\_id($id);

$data['title'] = "Dettaglio Giocatore";

$this->load->view('templates/header', $data);

$this->load->view('giocatore\_detail\_view', $data);

$this->load->view('templates/footer');

}

// Altri metodi: classifiche, head2head, biglietti

public function classifiche(){

// Implementa il recupero e la visualizzazione delle classifiche

$data['title'] = "Classifiche";

$this->load->view('templates/header', $data);

$this->load->view('classifiche\_view', $data);

$this->load->view('templates/footer');

}

public function head2head(){

// Implementa la logica per il confronto tra giocatori

$data['title'] = "Head2Head";

$this->load->view('templates/header', $data);

$this->load->view('head2head\_view', $data);

$this->load->view('templates/footer');

}

public function biglietti(){

// Implementa la visualizzazione e la logica d'acquisto biglietti

$data['title'] = "Biglietti";

$this->load->view('templates/header', $data);

$this->load->view('biglietti\_view', $data);

$this->load->view('templates/footer');

}

}

**Nota:** Per ogni pagina (classifiche, head2head, biglietti) potrai creare relativi modelli e viste analogamente a quanto mostrato.

**5. Creazione delle Viste (Views)**

**a. Template Header e Footer (navbar inclusa)**

**File:** application/views/templates/header.php

php

Copia

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title><?php echo isset($title) ? $title : 'Tennis Project'; ?></title>

<!-- Puoi includere qui CSS e JS -->

<link rel="stylesheet" href="<?php echo base\_url('assets/css/style.css'); ?>">

</head>

<body>

<nav>

<ul>

<li><a href="<?php echo site\_url('home'); ?>">Home</a></li>

<li><a href="<?php echo site\_url('home/partite'); ?>">Partite</a></li>

<li><a href="<?php echo site\_url('home/giocatori'); ?>">Giocatori</a></li>

<li><a href="<?php echo site\_url('home/classifiche'); ?>">Classifiche</a></li>

<li><a href="<?php echo site\_url('home/head2head'); ?>">Head2Head</a></li>

<li><a href="<?php echo site\_url('home/biglietti'); ?>">Biglietti</a></li>

<li><a href="<?php echo site\_url('login/logout'); ?>">Logout</a></li>

</ul>

</nav>

**File:** application/views/templates/footer.php

php

Copia

<footer>

<p>&copy; <?php echo date('Y'); ?> Tennis Project</p>

</footer>

</body>

</html>

**b. Vista Login**

**File:** application/views/login\_view.php

php

Copia

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Login - Tennis Project</title>

</head>

<body>

<h2>Login</h2>

<?php if($this->session->flashdata('error')): ?>

<p style="color:red;"><?php echo $this->session->flashdata('error'); ?></p>

<?php endif; ?>

<form action="<?php echo site\_url('login/process'); ?>" method="post">

<label>Email:</label>

<input type="email" name="email" required>

<br>

<label>Password:</label>

<input type="password" name="password" required>

<br>

<button type="submit">Accedi</button>

</form>

</body>

</html>

**c. Vista Home**

**File:** application/views/home\_view.php

php

Copia

<h1><?php echo $welcome\_message; ?></h1>

<p>

Benvenuto nel sito dedicato al Tennis. Qui potrai consultare le partite future, cercare informazioni sui giocatori, visualizzare le classifiche, effettuare confronti Head2Head e acquistare biglietti per i tornei.

</p>

**d. Vista Partite**

**File:** application/views/partite\_view.php

php

Copia

<h2>Partite Future</h2>

<?php if(!empty($partite)): ?>

<ul>

<?php foreach($partite as $partita): ?>

<li>

<strong>Torneo:</strong> <?php echo $partita->torneo\_nome; ?> -

<strong>Data:</strong> <?php echo $partita->data\_inizio; ?> -

<strong>Risultato:</strong> <?php echo $partita->risultato; ?>

</li>

<?php endforeach; ?>

</ul>

<?php else: ?>

<p>Nessuna partita futura trovata.</p>

<?php endif; ?>

**e. Vista Giocatori**

**File:** application/views/giocatori\_view.php

php

Copia

<h2>Ricerca Giocatori</h2>

<form method="get" action="<?php echo site\_url('home/giocatori'); ?>">

<input type="text" name="search" placeholder="Inserisci nome o cognome" value="<?php echo set\_value('search'); ?>">

<button type="submit">Cerca</button>

</form>

<?php if(!empty($giocatori)): ?>

<ul>

<?php foreach($giocatori as $giocatore): ?>

<li>

<?php echo $giocatore->nome . ' ' . $giocatore->cognome; ?>

<br>

<!-- Mostriamo una foto predefinita (in un caso reale il campo foto andrebbe aggiunto alla tabella) -->

<img src="<?php echo base\_url('assets/images/default\_player.jpg'); ?>" alt="<?php echo $giocatore->nome; ?>" width="100">

<br>

<a href="<?php echo site\_url('home/giocatore/' . $giocatore->id); ?>">Dettagli</a>

</li>

<?php endforeach; ?>

</ul>

<?php else: ?>

<p>Nessun giocatore trovato.</p>

<?php endif; ?>

**f. Vista Dettaglio Giocatore**

**File:** application/views/giocatore\_detail\_view.php

php

Copia

<h2>Dettaglio Giocatore</h2>

<?php if(!empty($giocatore)): ?>

<p><strong>Nome:</strong> <?php echo $giocatore->nome . ' ' . $giocatore->cognome; ?></p>

<p><strong>Nazione:</strong> <?php echo $giocatore->nazione; ?></p>

<p><strong>Età:</strong> <?php echo $giocatore->eta; ?></p>

<img src="<?php echo base\_url('assets/images/default\_player.jpg'); ?>" alt="<?php echo $giocatore->nome; ?>" width="150">

<?php else: ?>

<p>Giocatore non trovato.</p>

<?php endif; ?>

**6. Struttura Finale del Progetto**

La struttura delle cartelle del progetto CodeIgniter potrebbe essere simile a questa:

pgsql

Copia

tennis\_project/

├── application/

│ ├── controllers/

│ │ ├── Home.php

│ │ └── Login.php

│ ├── models/

│ │ ├── Utenti\_model.php

│ │ ├── Partite\_model.php

│ │ └── Giocatori\_model.php

│ └── views/

│ ├── templates/

│ │ ├── header.php

│ │ └── footer.php

│ ├── login\_view.php

│ ├── home\_view.php

│ ├── partite\_view.php

│ ├── giocatori\_view.php

│ └── giocatore\_detail\_view.php

├── assets/

│ ├── css/

│ │ └── style.css

│ └── images/

│ └── default\_player.jpg

└── system/

**7. Passi per la Costruzione del Sito**

1. **Preparazione del Database:**
   * Esegui lo script SQL fornito per creare il database “tennis” e popolare le tabelle con dati di esempio.
2. **Installazione di CodeIgniter:**
   * Scarica CodeIgniter, posizionalo nella directory del tuo server e configura config.php e database.php con le impostazioni corrette.
3. **Creazione dei Modelli:**
   * Crea i file dei modelli per gestire le operazioni sul database (utenti, partite, giocatori, ecc.).
4. **Creazione dei Controller:**
   * Realizza il controller Login per gestire l’autenticazione.
   * Realizza il controller Home per la home page e le varie sezioni (partite, giocatori, classifiche, head2head, biglietti).
   * Implementa nel controller eventuali metodi aggiuntivi per gestire le operazioni (es. visualizzazione dettagli giocatore, logout).
5. **Realizzazione delle Viste:**
   * Crea le viste per il form di login, la home page, e per ciascuna sezione (ad es. partite e giocatori).
   * Utilizza template comuni (header e footer) per avere una navbar omogenea su tutte le pagine.
6. **Test e Debug:**
   * Accedi alla pagina di login (es. http://localhost/tennis\_project/index.php/login), prova a effettuare l’accesso con le credenziali presenti nel database e verifica il corretto funzionamento della navigazione.
7. **Estensioni Future:**
   * Aggiungi le funzionalità per la pagina delle classifiche, per il confronto head2head e per l’acquisto dei biglietti creando ulteriori modelli, controller e viste.
   * Se necessario, amplia la tabella dei giocatori aggiungendo il campo “foto” e altri parametri per le statistiche.

Questo esempio rappresenta una solida base per realizzare il progetto richiesto. Naturalmente, in un ambiente di produzione andrebbero implementate ulteriori misure di sicurezza, validazioni e un sistema di gestione degli errori più completo. Buon lavoro nello sviluppo del tuo sito di simulazione tennis con CodeIgniter e PHP!