

# DOCUMENTATION TECHNIQUE

---

## Tennis

---

### **CFPT-Informatique**

### Tables des matières

- [1. Table des versions](#)
- [2. Introduction](#)
- [3. Résumé du cahier des charges](#)
  - [3.1. Organisation](#)
  - [3.2. Livrables](#)
  - [3.3. Matériel et logiciels à disposition](#)
  - [3.4. Description de l'application](#)
- [4. Méthodologie](#)
  - [4.1. S'informer](#)
  - [4.2. Planifier](#)
  - [4.3. Décider](#)
  - [4.4. Réaliser](#)
  - [4.5. Contrôler](#)
  - [4.6. Evaluer](#)
- [5. Planification](#)
  - [5.1. Product Backlog](#)
  - [5.2. Plan previsionnel](#)
- [6. Analyse Fonctionnelle](#)
  - [6.1. Fonctionnalités](#)
  - [6.2. Description des fonctionnalités](#)
  - [6.3. Mesure de sécurité](#)
- [7. Analyse Organique](#)
  - [7.0.1. Page index](#)
  - [7.0.2. Page Creation](#)
  - [7.0.3. Page Modification](#)
  - [7.0.4. Page Tournois](#)
  - [7.0.5. Page Joueur](#)
  - [7.1. Technologies utilisées](#)
  - [7.2. Environnement](#)
  - [7.3. Description de la base de données](#)
    - [7.3.1. Modèle logique de données](#)
- [8. Tests](#)
  - [8.1. Environnement des tests](#)
  - [8.2. Plan de test](#)
  - [8.3. Rapport de test](#)
- [9. Conclusion](#)

- [10. Bibliographie](#)
- [11. Annexes](#)
  - [11.1. Planning Prévisionnel](#)
  - [11.2. Planning Effectif](#)
  - [11.3. Code Source](#)

## 1. Table des versions

a faire !!!!!!!

## 2. Introduction

Ce document est un rapport montrant la conception du projet. Ce projet, réalisé dans le cadre du TPI (Travail Pratique Individuel), permet de valider mes compétences afin l'obtention du CFC. Le site web Tennis est un site web permettant la gestion de tournois de tennis. Le tournoi prend en compte plusieurs paramètres comme le type de tournoi, le nombre de set ou encore le genre (homme/femme) de joueur qui peut participer au tournoi.

## 3. Résumé du cahier des charges

Je dois réaliser un site web en adéquation avec le cahier des charges qui m'a été fourni

### 3.1. Organisation

Elève	Formateur	Expert	Expert
Bonvallat Dan	Bergeret Patrick Joseph	Strazzeri Mickaël	Vanini Daniel
<a href="mailto:dan.bnvll@eduge.ch">dan.bnvll@eduge.ch</a>	<a href="mailto:edu-bergeretpj@edu.ge.ch">edu-bergeretpj@edu.ge.ch</a>	<a href="mailto:mickael.strazzeri@git-it.ch">mickael.strazzeri@git-it.ch</a>	<a href="mailto:daniel.vanini@skyguide.ch">daniel.vanini@skyguide.ch</a>

### 3.2. Livrables

- Planning
- Documentation technique avec codes sources
- Manuel utilisateur
- Résumé du rapport du TPI
- Journal de bord

### 3.3. Matériel et logiciels à disposition

☐ Un PC standard école, 2 écrans ☐ Windows 10 ☐ EasyPHP, Laragon, WAMP ou autre ☐ mysql workbench, phpmyadmin ☐ Visual Studio Code ☐ Suite Office

### 3.4. Description de l'application

L'application permet de réaliser la gestion de tournois de tennis. Elle fournit une planification des matchs, l'inscription des joueurs et l'enregistrement des résultats des matchs. L'application se compose de deux parties :

- Gestion des tournois,
- Gestion des matchs.

Les fonctionnalités disponibles sont les suivantes :

- Créer, modifier, supprimer un tournoi,
- La liste des joueurs inscrits au tournoi est fournie,
- Planifier les matchs,
- Enregistrer les résultats des matchs,
- Effectuer des recherches : liste des joueurs, joueurs inscrits à un tournoi, liste des matchs, résultat, vainqueur, perdant d'un match, résultats des match, ...

Règles de gestion Tournois :

- La création est faite avant le début du tournoi,
- Le processus de création comporte quatre étapes : ◦ Initialisation avec les informations du tournoi ◦ Répartition des joueurs dans les tableaux haut et bas, ◦ Planification des matchs
- Après la date de début du tournoi le tournoi est automatiquement verrouillé,
- Le tournoi est aussi verrouillé sur demande explicite,
- Une fois verrouillé le tournoi ne peut plus être modifié ni supprimé,
- Copier un tournoi,
- Le tournoi ne prend en charge que des matchs en simple,
- Le genre est masculin ou féminin,
- Le nombre de sets est trois ou cinq
- La surface est : terre battue, gazon, surface rapide, ...
- La catégorie est : Grand Chelem, Masters, Masters 1000, ATP 500 Series, ATP 250 Series, ATP Challenge Tour, ...
- Tournoi avec ou sans jeu décisif.

Joueurs :

- La liste des joueurs inscrits au tournoi est fournie.

Matchs :

- Un match se joue en trois ou cinq sets (défini au niveau du tournoi)
- Le joueur qui gagne deux sets ou trois sets est déclaré vainqueur.
- Un match est gagné par forfait si un joueur abandonne en cours de match ou déclare forfait avant le début du match. Le score est celui au moment de l'abandon ou reste non rempli.

Sets :

- Dans un tournoi sans jeu décisif, le joueur qui gagne six jeux ou plus avec au moins deux jeux d'écart, remporte le set, exemples 6-1, 6-4, 7-5, 10-8, 14-12, ...
- Dans un tournoi avec jeu décisif, le joueur qui gagne six ou sept jeux avec deux jeux d'écart ou en cas d'égalité 6-6, gagne le jeu décisif, remporte le set, exemples 6-0, ..., 6-4, 7-5 ou 7-6

Jeux :

- Un jeu est remporté par le joueur qui arrive à quatre points ou plus avec au moins deux points d'écart,

- Un jeu décisif est gagné par le joueur qui arrive à six points minimum avec deux points d'écart.

#### Planification :

- Un tournoi est composé de deux tableaux : haut et bas,
- La répartition des joueurs se fait en fonction des têtes de série,
- Les têtes de série impaires sont placées dans le tableau haut,
- Les têtes de série paires sont placées dans le tableau bas,
- Le nombre de tours dans le tournoi dépend du nombre de participants, exemple pour 32 participants, 5 tours à jouer finale incluse (1er tour, 2e tour, 1/4, 1/2 et finale),
- Les quatre premières têtes de série ne se rencontrent qu'à partir des 1/4 de finales.

#### Interfaces

« menus » Tournois : liste, affiche, recherche, crée, modifie, supprime, copie, verrouille Joueurs : liste, affiche, recherche Matches : planifie, répartie les joueurs dans les tableaux, consulte Résultats : enregistre, modifie, efface, consulte

« accueil » La page d'accueil affiche par défaut le dernier tournoi consulté, avec ses informations. Une zone de recherche présente une liste déroulante avec les tournois, un clic sur un tournoi, met à jour l'affichage avec les informations du tournoi sélectionné.

« tournois » La page affiche la liste des tournois. Les actions disponibles : rechercher, voir, créer, modifier, supprimer, copier et verrouillé un tournoi.

« joueurs » La page affiche la liste des joueurs. Les actions disponibles : rechercher, voir un joueur.

« matchs » La page affiche la liste des matchs du tournoi sélectionné. Les actions disponibles : rechercher, planifier, répartir les joueurs et voir un match.

« résultats » La page affiche le score d'un match. Les actions disponibles : sélectionner, enregistrer, modifier et effacer le score d'un match.

## 4. Méthodologie

Afin de planifier mon projet, j'utilise la méthodologie en six étapes.

### 4.1. S'informer

La première chose que j'ai faite est de lire attentivement l'énoncé de mon tpi Après la lecture du tpi, j'ai appelé mon formateur afin de lui poser des questions sur l'énoncé.

### 4.2. Planifier

Au début du projet, j'ai découpé le travail que je devais faire pour savoir ce qui est le plus important, et savoir ce qui peut possiblement poser plus de problème. Pour tous les points de l'énoncé, j'ai fixé une priorité afin d'avoir un ordre d'importance pour réaliser les tâches que j'ai à faire. Les niveaux sont:

- B, Bloquant, X
- C, Critique, !
- I, Important, +

- S, Secondaire, -

Après avoir découpé mon travail, j'ai stocké chaque partie dans un product backlog. Une fois terminé, j'ai créé un planing prévisionnel afin d'avoir une ligne directrice.

### 4.3. Décider

Lors de la réalisation de mon projet j'ai couramment du prendre des décisions et faire des choix. Quand je dois faire un choix je réfléchis longuement afin de prendre la meilleur décision possible. Il arrive, des fois, que certains choix soient compliqués à prendre, alors, lorsque j'ai un doute j'en fais part à mon formateur.

### 4.4. Réaliser

Après avoir prit les décisions qui me semble juste, je continue sur le travail qui m'a posé problème précédemment (Implémentation de code ou rédaction de la documentation).

### 4.5. Contrôler

Dès qu'une fonctionnalité est terminée je vais immédiatement la tester dans plusieurs cas différents pour être sûr que la fonctionnalité fonctionne bien. Lorsque le site web est terminé, je teste l'ensemble des fonctionnalités du site.

### 4.6. Evaluer

Pour finir, j'applique cette dernière étape pour pouvoir savoir ce qui peut être améliorable. Grâce au journal de bord, où je note tout ce que je fais, je peux relire tout ce que j'ai fait la journée et donc voir où il y a des points améliorables.

## 5. Planification

### 5.1. Product Backlog

Nom	1: Créer un tournois
Description	En tant qu'utilisateur je peux créer un tournois en remplissant un formulaire
Priorité	B: Bloquant X
Nom	2: Rechercher un tournois
Description	En tant qu'utilisateur je peux rechercher un tournois via une barre de recherche
Priorité	S: Secondaire -
Nom	3: Modifier un tournois
Description	En tant qu'utilisateur je peux modifier les conditions d'un tournois après sa création
Priorité	S: Secondaire -
Nom	4: Supprimer un tournois
Description	En tant qu'utilisateur je peux supprimer un tournois

Nom	4: Supprimer un tournois
Priorité	I: Important +
Nom	5: Enregistrer les résultats des matches
Description	En tant qu'utilisateur je peux sauvgarder le résultat du match
Priorité	I: Important +
Nom	6: Planifier les matches
Description	En tant qu'utilisateur je peux planifier la date des matches
Priorité	I: Important +
Nom	7: Rechercher un joueur
Description	En tant qu'utilisateur je peux rechercher un joueur et savoir dans quel tournois il est inscrit
Priorité	I: Important +

5.2. Plan previsionnel

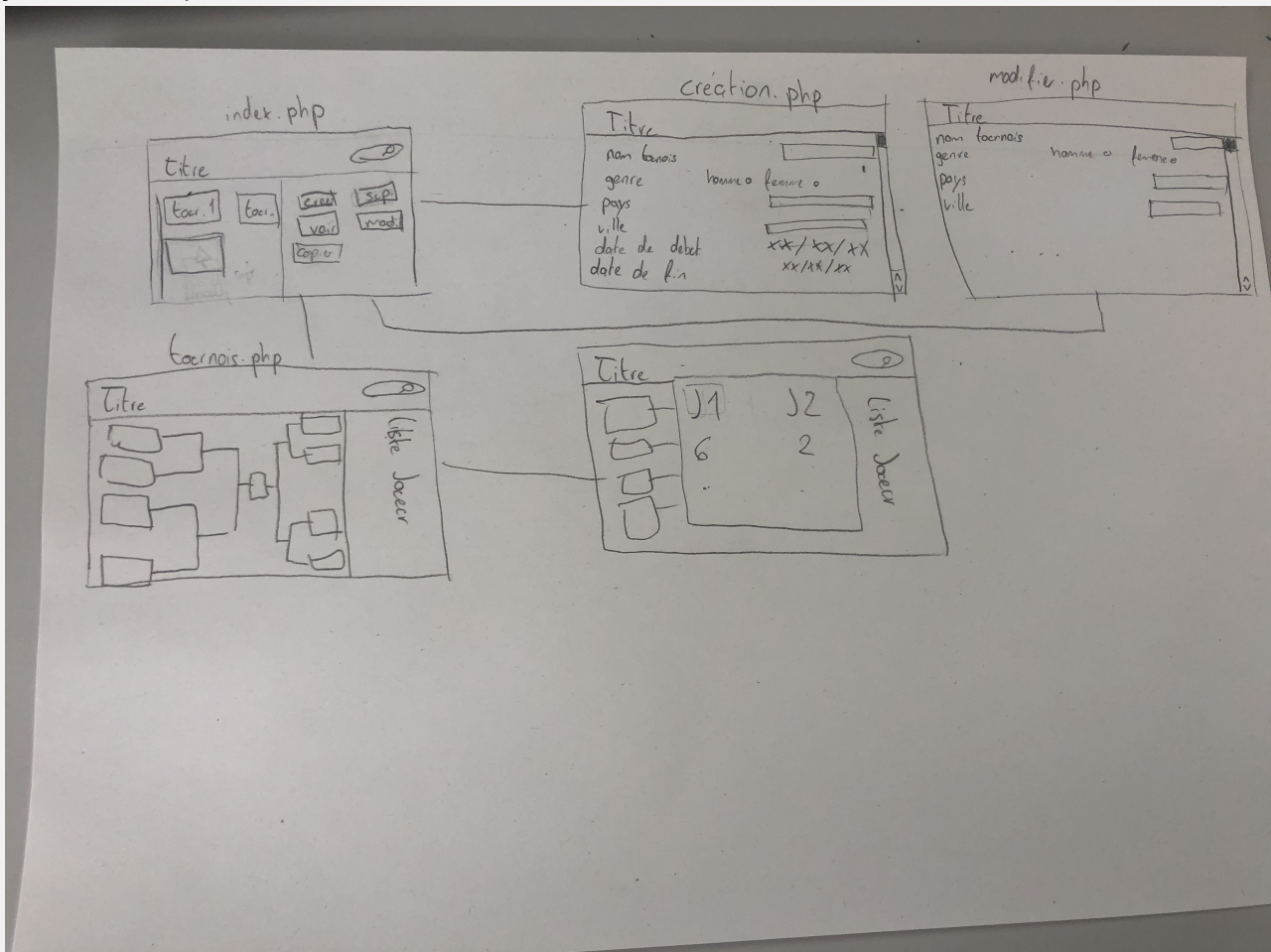
Voici le plan prévisionnel que j'ai réalisé

Tâches à réaliser	Temps nécessaire	1er jour	2e jour	3e jour	4e jour	5e jour	6e jour	7e jour	8e jour	9e jour	10e jour	11e jour	Total
Analyse / Mise en place de l'environement	04:00												00:00
Prise en connaissance du cahier des charges	01:00	01:00											01:00
Planification	01:00	01:00											01:00
Question	01:00	01:00											01:00
Création git/repository, extension markdown vscode, connection serveur + base de donn�	01:00	01:00											01:00
													00:00
													00:00
DEVELOPPEMENT	44:30												00:00
Cr�ation mod�le conceptuel	01:00		01:00										01:00
Devloppement page tournois	05:00		05:00										05:00
Impl�menter des joueurs	00:30					00:30:00							00:30
Ajout fonctionnalit�s cr�er trounois	05:00			05:00									05:00
Ajout fonctionnalit�s modifier trounois	07:00				05:00	02:00							07:00
Ajout fonctionnalit�s supprimer trounois	05:00			05:00									05:00
Planification des matches	05:00						05:00						05:00
Recherche matche	02:00							02:00					02:00
Recherche joueur	02:00						02:00						02:00
Recherche tournois	02:00						02:00						02:00
Test	10:00				03:00	04:00		03:00					10:00
													00:00
ADMINISTRATION	40:00												00:00
Documentation technique	19:00	02:00	02:00		02:00	02:00		03:00	02:00	02:00	02:00	02:00	19:00
Documentation utilisateur	10:00								03:00	03:00	02:00	02:00	10:00
Journal de Bord	11:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	01:00	11:00
													00:00
													00:00
													00:00
	88:30	07:00	09:00	11:00	11:00	09:30	10:00	09:00	06:00	06:00	05:00	05:00	88:30

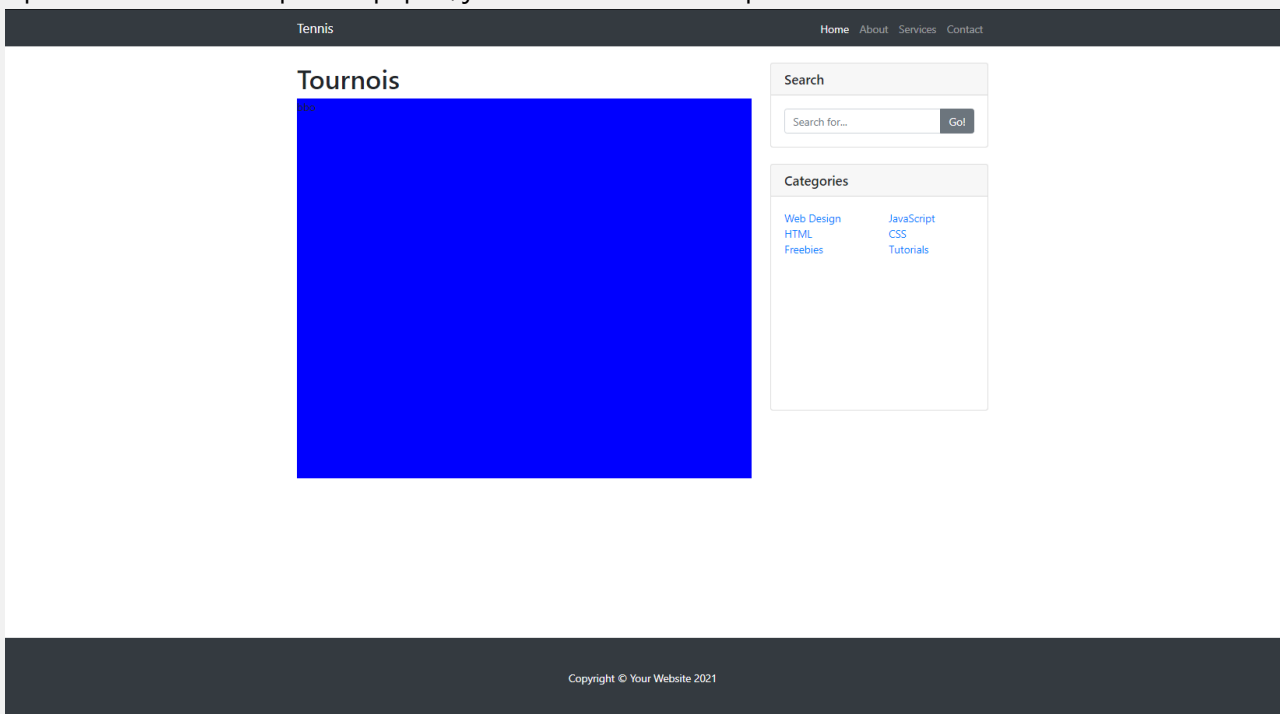
6. Analyse Fonctionnelle

Il y au totale 5 pages dans mon site web. Sur l'une d'entre elles (tournois.php) il y a un pop up qui fait apparaitre les 2 joueurs, l'utilisateur peut enregistrer les r sultats des 2 joueurs Pour r aliser ces pages,

j'ai fais un croquis



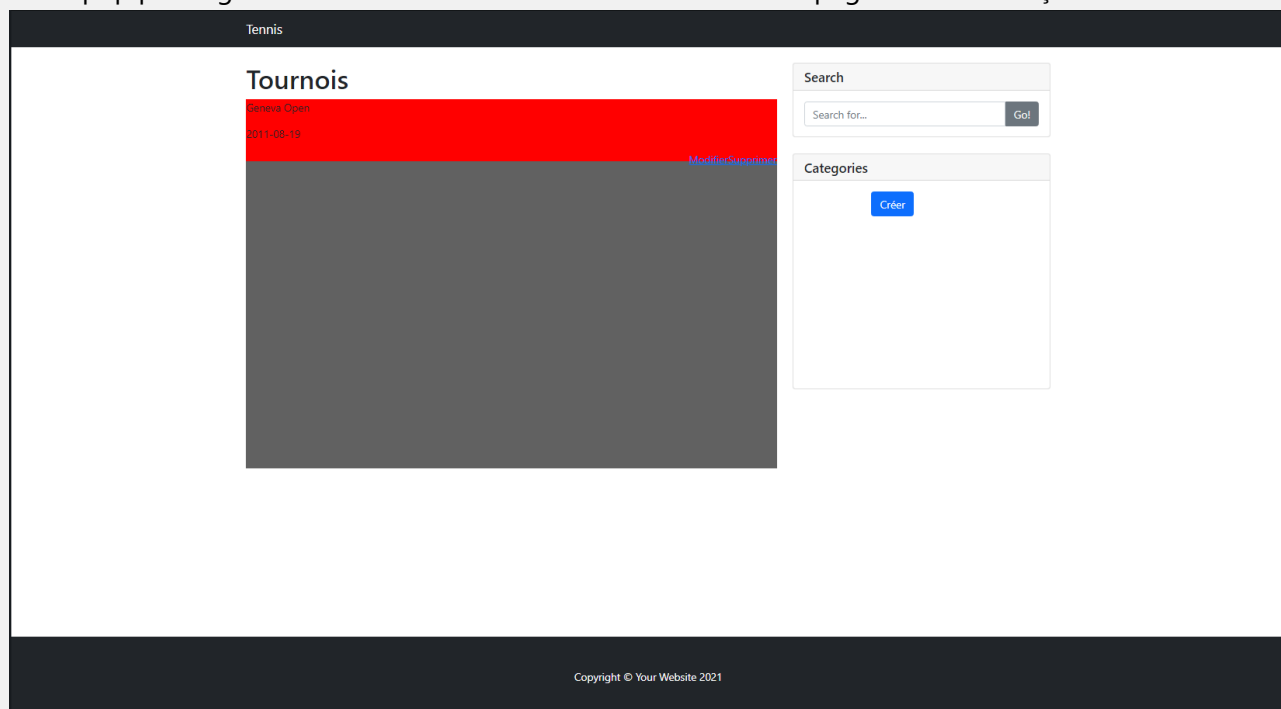
Après avoir fais le croquis sur papier, je decide de faire un squelette du site



Je commence alors par créer la page index.php. Cette page est un menu ou ce trouve les tournois créé par l'utilisateur. Je crée ensuite la page creation.php. Cette page est un formulaire ou l'utilisateur pourra créer un tournois.

## 6.1. Fonctionnalités

Sur la page index.php, l'utilisateur peut créer un tournoi en appuyant sur un bouton créer qui l'emmènera sur la page création.php. Une fois le tournoi créé, le tournoi s'affichera dans le compartiment en gris. À côté du tournoi, se trouvera un bouton pour supprimer le tournoi, un bouton modifier, un autre pour voir le tournoi et un dernier pour copier le tournoi. La page index.php peut également faire une recherche du tournoi. Cette page ressemble à ça



Sur la page creation.php, l'utilisateur a un formulaire avec plusieurs champs. Il doit remplir les champs pour créer le tournoi.



Tennis

Home About Services Contact

## Tournois

Nom du tournois  
Geneva Open

Pays  
Suisse

Ville  
Genève

Date de debut  
19.08.2011

Date de fin  
19.08.2011

☐ Homme  
☐ Femme

Genre

Dotation  
0

Nombre de joueurs  
16

Nombre de sets gagnants  
2

Type de terrains  
terre battue

Copyright © Your Website 2021

Tennis

Home About Services Contact

## Tournois

Ville  
Genève

Date de debut  
19.08.2011

Date de fin  
19.08.2011

☐ Homme  
☐ Femme

Genre

Dotation  
0

Nombre de joueurs  
16

Nombre de sets gagnants  
2

Type de terrains  
terre battue

type de tournois  
Grand Chelem

☐ Jeu décisif

Créer

Copyright © Your Website 2021

La page modification.php est pareil que la page creation.php. La différence est que lorsque l'utilisateur appuie sur le bouton modifié d'un tournoi, l'utilisateur se retrouvera sur une page pareil à la page creation.php, sauf que les données du tournoi sont déjà entrées dans les champs du formulaire, alors l'utilisateur a juste à modifier le champ qui lui convient pas.

The image displays two screenshots of a web application interface for managing tennis tournaments. Both screenshots show a form titled 'Tournois' with a dark header bar containing 'Tennis' and navigation links: 'Home', 'About', 'Services', and 'Contact'.

**Top Screenshot (Ajouter un tournoi):**

- Nom du tournoi:** Geneva Open
- Pays:** Suisse
- Ville:** Genève
- Date de debut:** 19.08.2011
- Date de fin:** 19.08.2011
- Genre:** ☐ Homme, ☒ Femme
- Dotation:** 0
- Nombre de joueurs:** 16
- Nombre de sets gagnants:** 2
- Type de terrains:** terre battue

**Bottom Screenshot (Modifier un tournoi):**

- Ville:** Genève
- Date de debut:** 19.08.2011
- Date de fin:** 19.08.2011
- Genre:** ☐ Homme, ☒ Femme
- Dotation:** 0
- Nombre de joueurs:** 16
- Nombre de sets gagnants:** 2
- Type de terrains:** terre battue
- type de tournoi:** Grand Chelem
- Jeu décisif:** ☐
- Modifier:** A blue button is visible at the bottom right of the form.

Lorsque l'utilisateur clique sur le bouton voir d'un tournoi sur la page index.php, il sera envoyé sur la page tournois.php. C'est sur cette page qu'aura lieu les matches et ou l'utilisateur entre le score des différents matches. METTRE IMAGE !!!!

Quand un utilisateur appuie sur le bouton supprimé, le tournoi est supprimé.

PAGE COPIER !!!

PAGE JOUEUR !!!

## 6.2. Description des fonctionnalités

Sur la page index l'utilisateur peut rechercher un tournoi. Il doit juste entré un mot dans la barre de recherche et à appuyer sur le bouton recherche et les tournois vont apparaîtres du plus au moins

pertinent. Il peut également appuyer sur les bouton créer, modifier, voir, copier et supprimer. Sur la page creation, l'utilisateur peut rentrer différentes données dans les champs du formulaire. Il peut ensuite cliquer sur le bouton créer qui va envoyer les données dans la base de données et l'utilisateur sera redirigé sur la page index. Sur la page modification.php l'utilisateur peut modifier un ou plusieurs des champs du formulaire. Une fois qu'il appuie sur le bouton modifier, les données seront envoyées dans la base de données et vont remplacer les anciennes données. L'utilisateur sera redirigé sur la page index.php. Lorsque l'utilisateur appuie sur le bouton supprimé, les données du tournoi vont être envoyées sur une page supprimer.php afin de supprimer les données de la base de données. L'utilisateur ne change pas de page. Lorsque l'utilisateur clique sur le bouton voir, il sera redirigé sur la page tournois.php. Sur cette page l'utilisateur pourra entrer les données des matches comme la date de la rencontre et les points des matches. Il pourra aussi télécharger la fiche du matche.

FAIRE PAGE COPIER !!!

### 6.3. Mesure de sécurité

Pour ce qui est de la sécurité, je filtre toujours les inputs avant de traiter les données. Lorsque je passe des données en get, je les filtre directement sur la page ou les données ont été envoyé.

## 7. Analyse Organique

### 7.0.1. Page index

Sur la page index il y a un lien qui permet de revenir a la page index. Le lien se trouve sur le mot "Tennis" qui est le titre de la page. Dans le lien la méthode de redirection est utilisé.

Sur la page index se trouve également un bouton créer qui redirige l'utilisateur sur la page creation.php. Lorsque l'utilisateur appuie sur le bouton, on vérifie que le bouton a été set puis on utilise une fonction avec un paramètre, qui est le chemin d'accès, qui permet de redirectionner l'utilisateur sur la page creation.

```
if (isset($_POST['creer'])) {  
    redirection($cheminCreer);  
}
```

```
function redirection($chemin)  
{  
    header("Location: $chemin");  
    exit();  
}
```

L'utilisateur peut chercher les tournois créés grâce à une barre de recherche ainsi que d'un bouton. A COMPLETER

Lorsque l'utilisateur supprime un tournoi, l'id du tournoi et l'id de la catégorie sont envoyés en get sur une page supprimé ou l'utilisateur n'a pas accès. Sur cette page les id sont d'abord filtrés puis entrés

dans une fonction afin de récupérer les données du tournoi créé et de la catégorie. Après que toutes les données aies été récupérer, deux fonctions vont êtres utilisées pour, d'abord, supprimer les données du tournoi puis ensuite supprimer les données de la catégorie. Les fonctions ont pour paramètre l'id du tournoi et l'id de la catégorie. Pour finir la page supprimer.php va se rediriger sur la page index avec la même fonction de redirection

```
$cheminSupprimer?idTournois=" . (int)$tournois["idTournois"] . "&idCategorie=" .  
(int)$tournois["idCategorie"]
```

```
$idTournois = filter_input(INPUT_GET, 'idTournois', FILTER_VALIDATE_INT);  
$idCategorie = filter_input(INPUT_GET, 'idCategorie', FILTER_VALIDATE_INT);
```

```
deleteTournois($idTournois);  
deleteCategorie($idCategorie);
```

```
function deleteTournois($idTournois)  
{  
    static $ps = null;  
    $sql = "DELETE FROM `tennis_tpi`.`tournois` WHERE (`idTournois` =  
:ID_TOURNOIS);";  
    if ($ps == null) {  
        $ps = tennis_database()->prepare($sql);  
    }  
    $answer = false;  
    try {  
        $ps->bindParam(':ID_TOURNOIS', intval($idTournois), PDO::PARAM_INT);  
        $ps->execute();  
        $answer = ($ps->rowCount() > 0);  
    } catch (PDOException $e) {  
        echo $e->getMessage();  
    }  
    return $answer;  
}
```

```
function deleteCategorie($idCategorie)  
{  
    static $ps = null;  
    $sql = "DELETE FROM `tennis_tpi`.`categories` WHERE (`idCategorie` =  
:ID_CATEGORIE);";  
    if ($ps == null) {  
        $ps = tennis_database()->prepare($sql);  
    }  
    $answer = false;
```

```

try {
    $ps->bindParam(':ID_CATEGORIE', intval($idCategorie), PDO::PARAM_INT);
    $ps->execute();
    $answer = ($ps->rowCount() > 0);
} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

```
redirection($cheminIndex);
```

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

Lorsque l'utilisateur modifie un tournoi, on passe en get l'id du tournoi sur la page modification.php et on redirige l'utilisateur grâce à un lien sur cette même page.

```
$cheminModification?idTournoi=" . (int)$tournois["idTournoi"]
```

Quand l'utilisateur accède à la page tournoi.php, on passe l'id du tournoi en get et on le redirige grâce à un lien sur la page tournoi.php.

```
$cheminVoir?idTournoi=" . (int)$tournois["idTournoi"]
```

### 7.0.2. Page Creation

Sur la page creation il y a un lien qui permet de revenir à la page index. Le lien se trouve sur le mot "Tennis" qui est le titre de la page. Dans le lien la méthode de redirection est utilisée.

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

Lorsque l'utilisateur appuie sur le bouton pour créer le tournoi après avoir rempli le formulaire, on vérifie d'abord que le bouton ait été activé puis les données sont filtrées et stockées dans des variables. Après que les données soient filtrées, des méthodes sont utilisées pour insérer les données dans la base de données. Les fonctions ont pour paramètre les variables dont les données ont été stocké précédemment. Après avoir insérer les données dans la base de données, l'utilisateur sera redirigé sur la page index à l'aide de la fonction de redirection.

```
if (isset($_POST['creer'])) {
}
```

```
$nomTournois = filter_input(INPUT_POST, 'nomTournois', FILTER_SANITIZE_STRING);
$nomPays = filter_input(INPUT_POST, 'nomPays', FILTER_SANITIZE_STRING);
$nomVille = filter_input(INPUT_POST, 'nomVille', FILTER_SANITIZE_STRING);
$dateDebut = filter_input(INPUT_POST, 'dateDebut', FILTER_SANITIZE_STRING);
$dateFin = filter_input(INPUT_POST, 'dateFin', FILTER_SANITIZE_STRING);
$genre = filter_input(INPUT_POST, 'genreTournois', FILTER_SANITIZE_STRING);
$dotation = filter_input(INPUT_POST, 'dotation', FILTER_VALIDATE_INT);
$nbJoueursFiltre = filter_input(INPUT_POST, 'nbJoueurs', FILTER_SANITIZE_STRING);
$nbSetsFiltre = filter_input(INPUT_POST, 'nbSets', FILTER_SANITIZE_STRING);
$jeuDecisif = filter_input(INPUT_POST, 'jeuDecisif', FILTER_SANITIZE_STRING);
    if ($jeuDecisif != 1)
    {
        $jeuDecisif = 0;
    }
$surface = filter_input(INPUT_POST, 'surface', FILTER_SANITIZE_STRING);
$typeTournois = filter_input(INPUT_POST, 'typeTournois', FILTER_SANITIZE_STRING);
```

```
insertCategorie($genre, $dotation, $surface, $typeTournois, $jeuDecisif,
$nbSetsFiltre, $nbJoueursFiltre);
$idCategorie = recupIdCategorie();
insertTournois($nomTournois, $nomPays, $nomVille, $dateDebut, $dateFin,
$idCategorie["idCategorie"]);
```

```
function insertCategorie($genre, $dotation, $surface, $typeTournois, $jeuDecisif,
$nbSet, $nbParticipant)
{
    static $ps = null;
    $sql = "INSERT INTO `tennis_tpi`.`categories` (`genre`, `dotation`, `idSurface`,
`idType`, `jeuDecisif`, `nbSet`, `nbParticipant`) ";
    $sql .= "VALUES (:GENRE, :DOTATION, :ID_SURFACE, :ID_TYPE, :JEU_DECISIF,
:Nb_Set, :Nb_Participant)";
    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
    $answer = false;
```

```

try {
    $ps->bindParam(':GENRE', $genre, PDO::PARAM_BOOL);
    $ps->bindParam(':DOTATION', $dotation, PDO::PARAM_INT);
    $ps->bindParam(':ID_SURFACE', $surface, PDO::PARAM_INT);
    $ps->bindParam(':ID_TYPE', $typeTournois, PDO::PARAM_INT);
    $ps->bindParam(':JEU_DECISIF', $jeuDecisif, PDO::PARAM_BOOL);
    $ps->bindParam(':NB_SET', $nbSet, PDO::PARAM_INT);
    $ps->bindParam(':NB_PARTICIPANT', $nbParticipant, PDO::PARAM_INT);

    $answer = $ps->execute();
} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

```

function recupIdCategorie()
{
    static $ps = null;
    $sql = 'SELECT idCategorie ';
    $sql .= 'FROM tennis_tpi.categories ';
    $sql .= 'ORDER BY idCategorie ';
    $sql .= 'DESC LIMIT 1';

    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
    $answer = false;
    try {
        if ($ps->execute())
            $answer = $ps->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }

    return $answer;
}

```

```

function insertTournois($nom, $pays, $ville, $dateDebut, $dateFin, $idCategorie)
{
    static $ps = null;
    $sql = "INSERT INTO `tennis_tpi`.`tournois` (`nom`, `pays`, `ville`,
`dateDebut`, `dateFin`, `idCategorie`) ";
    $sql .= "VALUES (:NOM, :PAYS, :VILLE, :DATE_DEBUT, :DATE_FIN, :ID_CATEGORIE)";
    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
    $answer = false;
    try {

```

```

$ps->bindParam(':NOM', $nom, PDO::PARAM_STR);
$ps->bindParam(':PAYS', $pays, PDO::PARAM_STR);
$ps->bindParam(':VILLE', $ville, PDO::PARAM_STR);
$ps->bindParam(':DATE_DEBUT', $dateDebut, PDO::PARAM_STR);
$ps->bindParam(':DATE_FIN', $dateFin, PDO::PARAM_STR);
$ps->bindParam(':ID_CATEGORIE', intval($idCategorie), PDO::PARAM_INT);

$answer = $ps->execute();
} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

### 7.0.3. Page Modification

Sur la page modification il y a un lien qui permet de revenir à la page index. Le lien se trouve sur le mot "Tennis" qui est le titre de la page. Dans le lien la méthode de redirection est utilisée.

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

Lorsque l'utilisateur appuie sur le bouton pour modifier le tournoi après avoir modifié le formulaire, on vérifie d'abord que le bouton ait été activé puis les données sont filtrées et stockées dans des variables. Après que les données soient filtrées, des méthodes sont utilisées pour modifier les données dans la base de données. Les fonctions ont pour paramètre les variables dont les données ont été stockées précédemment. Après avoir modifié les données dans la base de données, l'utilisateur sera redirigé sur la page index à l'aide de la fonction de redirection.

```

if (isset($_POST['modifier'])) {
}

```

```

//filtrage des inputs
$nomTournois = filter_input(INPUT_POST, 'nomTournois',

```



```

FILTER_SANITIZE_STRING);
    $nomPays = filter_input(INPUT_POST, 'nomPays', FILTER_SANITIZE_STRING);
    $nomVille = filter_input(INPUT_POST, 'nomVille', FILTER_SANITIZE_STRING);
    $dateDebut = filter_input(INPUT_POST, 'dateDebut', FILTER_SANITIZE_STRING);
    $dateFin = filter_input(INPUT_POST, 'dateFin', FILTER_SANITIZE_STRING);
    $genre = filter_input(INPUT_POST, 'genreTournois', FILTER_SANITIZE_STRING);
    $dotation = filter_input(INPUT_POST, 'dotation', FILTER_VALIDATE_INT);
    $nbJoueursFiltre = filter_input(INPUT_POST, 'nbJoueurs',
FILTER_SANITIZE_STRING);
    $nbSetsFiltre = filter_input(INPUT_POST, 'nbSets', FILTER_SANITIZE_STRING);
    $jeuDecisif = filter_input(INPUT_POST, 'jeuDecisif', FILTER_SANITIZE_STRING);
    if ($jeuDecisif != 1)
    {
        $jeuDecisif = 0;
    }
    $surface = filter_input(INPUT_POST, 'surface', FILTER_SANITIZE_STRING);
    $typeTournois = filter_input(INPUT_POST, 'typeTournois',
FILTER_SANITIZE_STRING);

```

```

$idCategorie = recupIdCategorie();
updateCategorie($genre, $dotation, $surface, $typeTournois, $jeuDecisif,
$nbSetsFiltre, $nbJoueursFiltre, $idCategorie['idCategorie']);
updateTournois($nomTournois, $nomPays, $nomVille, $dateDebut, $dateFin,
$idCategorie['idCategorie'], $idTournois);

```

```

function recupIdCategorie()
{
    static $ps = null;
    $sql = 'SELECT idCategorie ';
    $sql .= 'FROM tennis_tpi.categories ';
    $sql .= 'ORDER BY idCategorie ';
    $sql .= 'DESC LIMIT 1';

    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
    $answer = false;
    try {
        if ($ps->execute())
            $answer = $ps->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }

    return $answer;
}

```

```

function updateCategorie($genre, $dotation, $idSurface, $idType, $jeuDecisif,
$nbSet, $nbParticipant, $idCategorie)
{
    static $ps = null;

    $sql = "UPDATE `tennis_tpi`.`categories` SET ";
    $sql .= "`genre` = :GENRE, ";
    $sql .= "`dotation` = :DOTATION, ";
    $sql .= "`idSurface` = :ID_SURFACE, ";
    $sql .= "`idType` = :ID_TYPE, ";
    $sql .= "`jeuDecisif` = :JEU_DECISIF, ";
    $sql .= "`nbSet` = :NB_SET, ";
    $sql .= "`nbParticipant` = :NB_PARTICIPANT ";
    $sql .= "WHERE (`idCategorie` = :ID_CATEGORIE)";
    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
    $answer = false;
    try {
        $ps->bindParam(':GENRE', intval($genre), PDO::PARAM_INT);
        $ps->bindParam(':DOTATION', intval($dotation), PDO::PARAM_INT);
        $ps->bindParam(':ID_SURFACE', intval($idSurface), PDO::PARAM_INT);
        $ps->bindParam(':ID_TYPE', intval($idType), PDO::PARAM_INT);
        $ps->bindParam(':JEU_DECISIF', intval($jeuDecisif), PDO::PARAM_INT);
        $ps->bindParam(':NB_SET', intval($nbSet), PDO::PARAM_INT);
        $ps->bindParam(':NB_PARTICIPANT', intval($nbParticipant), PDO::PARAM_INT);
        $ps->bindParam(':ID_CATEGORIE', intval($idCategorie), PDO::PARAM_INT);
        $ps->execute();
        $answer = ($ps->rowCount() > 0);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }
    return $answer;
}

```

```

function updateTournois($nom, $pays, $ville, $dateDebut, $dateFin, $idCategorie,
$idTournois)
{
    static $ps = null;

    $sql = "UPDATE `tennis_tpi`.`tournois` SET ";
    $sql .= "`nom` = :NOM, ";
    $sql .= "`pays` = :PAYS, ";
    $sql .= "`ville` = :VILLE, ";
    $sql .= "`dateDebut` = :DATE_DEBUT, ";
    $sql .= "`dateFin` = :DATE_FIN, ";
    $sql .= "`idCategorie` = :ID_CATEGORIE ";
    $sql .= "WHERE (`idTournois` = :ID_TOURNOIS)";
    if ($ps == null) {
        $ps = tennis_database()->prepare($sql);
    }
}

```

```

$answer = false;
try {
    $ps->bindParam(':NOM', $nom, PDO::PARAM_STR);
    $ps->bindParam(':PAYS', $pays, PDO::PARAM_STR);
    $ps->bindParam(':VILLE', $ville, PDO::PARAM_STR);
    $ps->bindParam(':DATE_DEBUT', $dateDebut, PDO::PARAM_STR);
    $ps->bindParam(':DATE_FIN', $dateFin, PDO::PARAM_STR);
    $ps->bindParam(':ID_CATEGORIE', intval($idCategorie), PDO::PARAM_INT);
    $ps->bindParam(':ID_TOURNOIS', intval($idTournois), PDO::PARAM_INT);
    $ps->execute();
    $answer = ($ps->rowCount() > 0);
} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

#### 7.0.4. Page Tournois

Sur la page tournois il y a un lien qui permet de revenir a la page index. Le lien ce trouve sur le mot "Tennis" qui est le titre de la page. Dans le lien la méthode de redirection est utilisé.

```

function redirection($chemin)
{
    header("Location: $chemin");
    exit();
}

```

Pour chaque match, l'utiisateur peut entrer la date du match, l'heure du match, sur quel terrain ce déroule le match et le résultat des sets. Après avoir enregistrer les informations du match, l'utilisateur peut télécharger les données du match en pdf.

#### 7.0.5. Page Joueur

Sur la page tournois il y a un lien qui permet de revenir a la page index. Le lien ce trouve sur le mot "Tennis" qui est le titre de la page. Dans le lien la méthode de redirection est utilisé.

```

function redirection($chemin)
{
    header("Location: $chemin");
}

```

```
exit();  
}
```

A COMPLETER !!!

## 7.1. Technologies utilisées

Les technologies qui sont utilisé dans ce projet sont:

- php Le php est principalement utilisé pour traiter les données, sécuriser le site et faire les traitements avec la base de données
- css Le css est utiliser pour décorer le site web mais aussi pour faciliter l'utilisation de l'utilisateur lorsqu'il navigue sur le site.
- html L'html permet de construire le site et de permettre certaines actions qui nécessite du php
- sql L'sql permet de traiter les informations qu'il reçoit dans la base de données et permet l'accès au php afin de permettre aux informations de circuler entre le site web et la base de données.
- markdown Le markdown est très lisible et très graphique, rédaction rapide. Utilisé pour rédiger des documents.
- debian Debian est une interface linux sous lequel il peut héberger un serveur apache
- apache2 Apache2 est un serveur qui permet de faire le lien entre la base de donnée et le site web
- wsl Wsl est une couche de compatibilité permettant d'exécuter des exécutable Linux sur Windows

## 7.2. Environnement

Pour la plupart des technologies utilisé, j'utilise Visual Studio Code. Visual Studio Code est un éditeur de code développé par Microsoft pour Windows. La communauté développe peut développer des extensions qui permet de coder de manière plus ergonomique. J'utilise Visual Studio Code pour le php, le css, l'html, markdown et wsl. Pour le sql j'utilise le programme DBeaver. Pour le serveur j'utilise apache2 sur l'interface Linux Debian.

## 7.3. Description de la base de données

La base de données compte 12 tables dont deux tables qui contiennent juste les id deux autres tables. Les tables sont :

- matches Contient les informations des matches, lié avec la table tournois, tours, scores, joue, terrains
- tournois Contient les informations du tournoi comme son nom ou encore l'endroit où il se trouve. Lié avec la table matches et categories.
- categories Contient les informations concernant les catégories des tournois comme le genre, le nombre de sets gagnant ou la dotation. Lié avec la table tournois, surfaces et types
- surfaces Contient le type de surface d'un terrain de tennis. Lié avec la table catégorie
- types Contient le type du tournoi (Grand Chelem, master 1000, ...). Lié avec la table categories.
- tours Contient le type de tours (demi-finale, finale, ...). Lié avec la table matches
- terrains Contient l'emplacement des terrains et leur nom comme terrainA - Nord. Lié avec la table matches

- scores Contient les id de la tables matches et la table sets. Lié avec la table sets et matches.
- sets Contient le score d'un set. Lié avec la table scores et la table jeu\_decisif
- jeu\_decisif Contient les points du jeu decisif. Lié avec la table set
- joue Contient les id de la table matches et joueurs. Lié avec la table matches et joueurs.
- joueurs Contient les informations des joueurs du tournoi. Lié avec la table joue.

### 7.3.1. Modèle logique de données

## 8. Tests

### 8.1. Environnement des tests

### 8.2. Plan de test

N°	Description du test	Résultat attendu
1	Test magnifique cil est vraiment trop bien ce test	Il se passe ça
2		

### 8.3. Rapport de test

OK -> le test fonctionne / NOK -> le test ne fonctionne pas

N°	Résultat obtenu	Validation
1	Test magnifique cil est vraiment trop bien ce test	OK (26.04.2021)
2		

## 9. Conclusion

## 10. Bibliographie

## 11. Annexes

### 11.1. Planning Prévisionnel

### 11.2. Planning Effectif

### 11.3. Code Source