



VINCI COURT

TENNIS CLUB MANAGEMENT WEBSITE

BINV1110 - Projet web - 2024

TABLE OF CONTENT

Table of Content	1
I. Context.....	3
II. Functional Requirements.....	3
II.1 Shared Pages	4
Home page.....	4
Navigation bar.....	4
Login page.....	5
Register page	5
II.2 Tournaments pages.....	5
Tournament list page.....	5
Tournament details page.....	5
Tournament creation page	6
II.3 Courts pages	6
Court list page	6
Court details page	6
Court bookings page	6
Court update page	7
II.4 Coaches pages	7
Coach list page	7
Coach bio page.....	7
Coach messages page	7
Coach update page	8
II.5 Database requirements.....	8
III. Non-functional requirements	9
Language	9
Database.....	9
Development.....	9

IV.	Project Management.....	9
	Organization and project tasks	9
	Sessions and meetings	10
	GitHub	10
V.	Project Plan	11
VI.	Deliverables	12
	Intermediate Submission 1	12
	Prototypes.....	12
	Routing table.....	12
	Submission	12
	Intermediate Submission 2	12
	DSD diagram	12
	SQL script	12
	Submission	13
	Final submission	13
	Report	13
	Code	13
	Submission	13
VII.	Attendance	13
VIII.	Evaluation	14

I. CONTEXT

Vinci Court is a bustling local tennis club, nestled within your community—a haven for tennis aficionados of all ages and skill levels. As the heartbeat of our neighbourhood's sporting scene, Vinci Court aspires to modernize its operations and elevate member experiences through a digital transformation. Your team has been summoned for a remarkable endeavour: to craft an innovative management website for Vinci Court Tennis Club.

In this ambitious initiative, you are tasked with laying the foundation for a sophisticated platform that will revolutionize how Vinci Court operates and engages with its members. This project marks the inception of what could be the next frontier in tennis club management. Prepare to embark on a journey that will shape the future of Vinci Court Tennis Club and redefine the standards of community-centric sports management.

II. FUNCTIONAL REQUIREMENTS

The website will allow any user (whether connected or not) to:

- Get information about Vinci Court.
- Create an account and login.
- Log in and logout of the site.
- Consult the next tournaments.
- Consult the details of the tennis courts.
- Read the bios of the coaches.

In addition to this, connected users are also allowed to:

- Register for a tournament.
- Unregister from a tournament.
- Book a tennis court.
- Consult their bookings.
- Manage their bookings.
- Send a message to a coach.
- Consult their messages and the responses.

Some connected users have additional rights as coaches of the club, and are allowed to:

- Create a tournament.
- Manage tennis courts.
- Manage their bio page.
- Consult their received messages and respond.

II.1 SHARED PAGES

HOME PAGE

This page presents the club. The user should see a description of the club (you are free here to make up the details and add your distinctive touch).

Below this description, users should see a few stats about the club:

- Number of upcoming tournaments.
- Number of upcoming court bookings.
- Number of messages sent in the last week.

At the bottom of the page, the user should see a presentation of your team, with your group number and each of your names.

NAVIGATION BAR

On top of all pages, the user should see a navigation bar that allows them to navigate to the different pages of the website. When the user is not connected, it should display the following links:

- Home page.
- Tournament list page.
- Court list page.
- Coach list page.
- Login page.
- Register page.

When the user is connected, the navigation bar should display the following links:

- Home page.
- Tournament list page.
- Court list page.
- Courts bookings page.
- Coach list page.
- Coach messages page.
- Logout.

When the user is connected and registered as a coach, it should display the following links:

- Home page.
- Tournament list page.
- Tournament creation page.
- Court list page.
- Courts bookings page.
- Coach list page.
- Coach messages page.
- Logout.

LOGIN PAGE

This page displays a form, with fields for an email and a password.

When the user submits the form and the credentials are correct, they are logged into the website and redirected to the home page. If at least one of the details is not right, the page displays an error message describing the problem instead.

REGISTER PAGE

This page displays a form, with fields for the user's first name, surname, email, password, and its confirmation.

When the user submits the form and the information are valid, their account is created, and they are redirected to the login page. If one field or more could not be validated, the page displays an error message describing the problem instead.

The email must end with the domain "vinci.be" (or any subdomain such as "student.vinci.be" or "coach.vinci.be"). The email must also be rejected if an account already exists with this email.

The password must be at least 8 characters long, and contain at least one lowercase and uppercase letter, and one digit number. The password confirmation must match the password.

Any new user registered on the site is a regular user and not a coach.

II.2 TOURNEMENTS PAGES

TOURNEMENT LIST PAGE

This page presents a list of all future tournaments, ordered by date. For each tournament, the page displays its name, date, and maximum number of participants.

When the user clicks on the name of a tournament, they are redirected to details page of this tournament.

TOURNAMENT DETAILS PAGE

This page displays all information about a tournament, with its banner image, name, date, the name of the coach that created it, the number of registered person and the maximal number of participants.

When the user asks for a tournament that does not exist, the page displays an error message instead.

If the user is connected and if 1) the tournament date has not happened yet, 2) the maximum number of participants has not been reached yet, and 3) the user is not already registered to this tournament, the page displays a button to register to this tournament.

When the user clicks on this button, they are registered to this tournament and the page is displayed again.

If the user is already registered to this tournament, the page displays a button to unregister from this tournament. When the user clicks on this button, they are unregistered, and the page is displayed again.

TOURNAMENT CREATION PAGE

This page displays a form, with fields for the tournament's name, banner image file, date, and maximum number of participants.

All fields are required. The date must be in the future. The maximum number of participants must be an integer greater than one. No tournament name must be unique.

When the user submits the form, the tournament is created, and the user is redirected to the tournaments page. If one of the fields is empty or not correct, the page displays an error instead.

If the user is not connected or not a coach, they are redirected to the tournament list page.

II.3 COURTS PAGES

COURT LIST PAGE

This page presents a list of all tennis courts, ordered by name. For each court, the page displays its name, and flooring type.

When the user clicks on the name of a tennis court, they are redirected to details page of this court.

If the user is connected as a coach, the list displays a button for each tennis court. When the user clicks on one of these buttons, they are redirected to the court modification page for this court.

COURT DETAILS PAGE

This page displays all information about a tennis court, with its name, flooring type, location, and picture.

If the user asks for a tennis court that does not exist, the page displays an error message instead.

If the user is connected, the page also displays a form to book the court. The form displays a list of all days of the following week.

When the user submits the form, if the court was already booked for that day, the page displays an error message. Otherwise, the court is booked by the user for the day they selected, and they are redirected to the court bookings page.

COURT BOOKINGS PAGE

This page displays a list of all courts that the user has booked in the future, ordered by booking date. For each court, the page displays its name, the booking date, and a button to unbook this court.

When the user clicks on one of these buttons, the booking is removed, and the page is displayed again.

If the user is not connected, they are redirected to the court list page.

COURT UPDATE PAGE

This page displays a form with fields for the court's name, flooring type, location, and picture file. When the page is first displayed, the fields are filled with the court's previous information.

All fields are required. The accepted flooring types are "grass", "clay", and "artificial". The form must show a dropdown menu where the user can select one of these values for the flooring type field. The location is a short textual description of the place where the court is located. The court name and location must be unique.

When the user submits the form, the court information is updated, and they are redirected to the court details page.

If the user asks for a tennis court that does not exist, the page displays an error message instead.

If the user is not connected or not a coach, they are redirected to the court details page.

II.4 COACHES PAGES

COACH LIST PAGE

This page displays a list of all coaches ordered by name. For each coach, the page displays their full name and email.

When the user clicks on the name of a coach, they are redirected to the bio page of this coach.

COACH BIO PAGE

This page displays all information about a coach, with their first name, surname, email, biography, and picture.

If the user is connected and not a coach, the page displays a form to send a message to this coach. When the user submits the form, the message is sent, and they are redirected to the messages page.

If the user **is the coach**, he is then logged on his own web page. The page displays a button to update its bio. When the user clicks on the update button, they are redirected to the update bio page.

If the user asks for a user that does not exist or is not a coach, the page displays an error message instead.

COACH MESSAGES PAGE

This page displays a list of messages, ordered from most recent to oldest.

There is no *conversation*, aka thread of messages on this page. Only a message and a single response. Messages are always initiated by regular users.

Any regular user can only see his own messages. For each message, the page displays the message and its response if it exists, or an indication the coach hasn't responded yet.

Any coach user can only see messages sent to him and he hasn't replied yet. For each message, the page displays the text, the name of sender, the date and time of sending. A form to send a response (with a "send" button) is

presented for each message. When the send button is clicked, the message is sent, and the page is refreshed. The replied message should not be displayed.

If the user is not connected, they are redirected to the coach list page.

COACH UPDATE PAGE

This page displays a form, with fields for the coach's biography, and picture file. When the page is first displayed, the fields are filled with their previous information. All fields are optional.

When the user submits the form, their information is updated, and they are redirected to their bio page.

If the user is not connected or not a coach, they are redirected to the coach list page.

II.5 DATABASE REQUIREMENTS

- Each tournament must have a unique name. Two tournaments can happen at the same date. As a simplification, a tournament is considered to always happen at a single date and not span multiple days.
- Each user must have a unique email.
- Each tennis court must have unique name and location.
- A message is sent from a user to another user. Users can send and receive multiple messages.
- A tournament is created by a user. Users can create multiple tournaments.
- A tennis court booking is made by a user for a tennis court. A user can place multiple bookings, and a tennis court can be booked multiple times. Only one user can book a tennis court at a certain date.
- A tournament registration is made by a user for a tournament. A user can make multiple registrations, and a tournament can register multiple users.

III. NON-FUNCTIONAL REQUIREMENTS

LANGUAGE

All your code (variables, routers, views, ...) and comments **must be in English!** The user interface language can be French or English. You should however keep the same language throughout the website.

Some names in this document are presented in English and should be visible in the application, such as the court's flooring types for instance. If you decide to have your interface in French, you should translate this information in French to be consistent.

DATABASE

- The database of the web site is an SQLite file.
- The database file must be created using the provided installation script. Details about the installation of the project will be provided later.

DEVELOPMENT

- The development language of the server is JavaScript.
- The recommended development IDE is Visual Studio Code.
- The design must follow the MVC architecture.
- The name of the routes and models shall respect rules and convention seen in the course: "BINV1051-1 Développement web: bases".
- The HTML code must be valid HTML 5. You should not include javascript in your HTML pages.
- The database design shall respect the rules and conventions seen in the course: "Gestion des données: bases".
- We recommend the use of the Bootstrap framework for HTML design and CSS. It produces friendly html pages with less effort.
- The folders' names containing the different parts should correspond to these used during the "Développement web : bases" classes.

IV. PROJECT MANAGEMENT

ORGANIZATION AND PROJECT TASKS

The project is scheduled for 6 weeks, with three sessions per week. The website targets many different pages, each with different data and expectations. Therefore, we split the tasks in the project into three parts based on the different pages. The three parts of the project are not completely independent from each other. For this reason, we created another part that we call the "Shared tasks".

- **Shared tasks:** Home page, Login page, Register page, Navigation.
- **Part A:** Tournament list page, tournament details page, tournament creation page.
- **Part B:** Court list page, court details page, court bookings page, court update page.
- **Part C:** Coach list page, coach bio page, coach messages page, coach update page.

You will work in a group of **three developers**. Each developer is in full charge of tasks in one part, and they have a joint responsibility for shared tasks. During the first session, you must decide who is working on what part of the project. Understanding every part of the project by each developer is important for successful implementation. However, evaluation is based on the tasks assigned to each developer.

The project template based on which you will start your work contains a basic implementation of the shared tasks pages. However, you will have to improve them to respect all the requirements of these pages.

Your project management method borrows from the SCRUM mindset. The project is split into small sets of features. Features are prioritized and split on a feasible one-week long plan (a sprint). Communication among different roles is an essential practice in SCRUM. A 5-minute discussion per day avoids many problems in the project. Therefore, communicate on the project with your partners and the professors: what is done and what is next.

SESSIONS AND MEETINGS

During the sessions, the professors will provide help on all parts of the project. Feel free to contact them to discuss your ideas and get their feedback.

GITHUB

- You will use GitHub as the source control tool for your project's code.
- **Check the tutorial on Moodle on using git.**
- **Follow the instructions** on Moodle on how to create your account and create your repository.

V. PROJECT PLAN

The project is scheduled to last for 6 weeks, from Week 7 to Week 12. There will be two compulsory intermediate submissions. Apart from these, you are free to organize your work as you prefer. We strongly recommend, however, that you follow the project schedule proposed below.

Week	Responsibility	Task
Week 7	Shared	Read and understand the assignment statement.
	Shared	Decide who will do part A, B or C.
	Shared	Prototypes for pages. See intermediate submission 1.
	Shared	Chart of every route. See intermediate submission 1.
22/03 20:00	Shared	Intermediate Submission 1.
Week 8	Shared	Draw the DSD of the database. See intermediate submission 2.
	Shared	Create a database creation script. See intermediate submission 2.
29/03 20:00	Shared	Intermediate Submission 2.
Week 9	Shared	Read and understand the corrections of previous submissions.
	Shared	Install the project. See instructions on Moodle.
	Shared	Create the design, global CSS, logo, and spirit of the website.
	Shared	Implement the home page.
	Shared	Implement the navigation.
	Shared	Implement the login page.
	Shared	Implement the register page.
Week 10	Part A	Implement the tournament list page.
	Part A	Implement the tournament details page.
	Part B	Implement the court list page.
	Part B	Implement the court details page.
	Part C	Implement the coach list page.
	Part C	Implement the coach bio page.
Week 11	Part A	Implement the tournament creation page.
	Part B	Implement the court bookings page.
	Part B	Implement the court update page.
	Part C	Implement the coach messages page.
	Part C	Implement the coach update page.
	Shared	Test that the different part work together
Week 12	Shared	Finalize the project.
	Shared	Test the interview scenarios.
	Shared	Make the report.
25/04 20:00	Shared	Final submission.

VI. DELIVERABLES

INTERMEDIATE SUBMISSION 1

In the first submission, you will analyse the requirements of the website and develop your interface.

PROTOTYPES

Create prototypes of the user interface for the following pages: Tournament details page, court details page, and coach bio page. These prototypes should be images reflecting on the content of your interface.

We advise you to use the website app.diagrams.net (formerly known as draw.io) as a tool to draw your prototypes. Similar other tools exist. You should export your prototypes as PDF files.

ROUTING TABLE

Create a table of all the routes of your program, describing their method, path, router, model, and view with a textual description of the purpose of this route. You can use the file *exoplanets_routetable_sample.docx* available on Moodle as an example for your own route table.

SUBMISSION

Export your prototypes as PDF files and bundle them in a folder called *prototypes_group_XX*, where *XX* is the number of the group you were assigned to. Name your route table file *routes_group_XX.docx*.

Put both this folder and file in another folder called *submission1_group_XX*. Zip the folder. The name of the zipped folder shall follow the same naming convention.

Upload the zipped file to the Moodle assignment called “Intermediate Submission 1” **before 22/03 20:00**. Only one student for each group must do this submission.

INTERMEDIATE SUBMISSION 2

In the second submission, you will create the structure of your database for the project.

DSD DIAGRAM

Draw the DSD diagram of your database, following the conventions seen in the course “Gestion des données : bases”. Use once again the website app.diagrams.net to create your diagram. You should export your diagram as a PDF file.

SQL SCRIPT

Create an SQL script that creates or re-creates the tables in your database. Export this script as an SQL file.

SUBMISSION

Name your DSD diagram *dsd_group_XX.pdf* where *XX* is the number of the group you were assigned to, and your SQL script *script_group_XX.sql*.

Put both files in a folder called *submission2_group_XX*. Zip the folder. The name of the zipped folder shall follow the same naming convention.

Upload the zipped file to the Moodle assignment called “Intermediate Submission 2” **before 29/03 20:00**. Only one student for each group must do this submission.

FINAL SUBMISSION

REPORT

For the final submission, you will submit a report presenting your project and the results you achieved. A template for the report is available on Moodle. The report language is **French** or **English**.

CODE

Your project template includes a “prepare_submission.sh” file. This script checks that your code is functional and creates the archive for submission. Launch this script (inside a terminal located at your project’s folder, run the command “bash prepare_submission.sh”). If your code is not functional, this script will display an error message. Otherwise, the script creates a submission archive called *project_group_XX_FIRSTNAME_SURNAME.zip*.

SUBMISSION

Name the report as: *rapport_group_XX.docx*. Upload this report to the Moodle assignment called “Final submission: Project report” **before 26/04 20:00**. Only one student for each group must do this submission.

Rename the submission archive with your group number, firstname and surname. Upload this zipped folder to the Moodle assignment called “Final submission: Project code” **before 26/04 20:00**. **Every student must do this submission**. If you have not correctly completed this submission, we will consider you as having not participated to the project, and you will not be able to participate to the interview during the exam session.

VII. ATTENDANCE

Attendance to the practical sessions is very important and is expected of every member at each session. We will especially pay attention to attendance during the first two weeks of the project. **Students absent without justification during this period may be removed from the project and assigned a null grade.**

Throughout the project, if one of your teammates does not participate properly and/or does not answer your calls, we ask you to contact a teacher during the practical sessions **as soon as possible** so we can help resolve the situation.

VIII. EVALUATION

The evaluation is split between two parts, the continuous evaluation regarding the assignments during the project and the interview during the examination session. You will be evaluated as a group for the continuous evaluation, and independently during the interview.

The interview during the examination session assesses whether the developer achieved their tasks correctly and by themselves. A professor will ask the developer to run test scenarios and to implement a change to the code to proof ownership.

- | | | |
|---|------------|-----|
| • Continuous evaluation | 50% | |
| ○ Intermediate submission 1 | | 20% |
| ○ Intermediate submission 2 | | 20% |
| ○ Report | | 10% |
| • Interview during the examination session | 50% | |