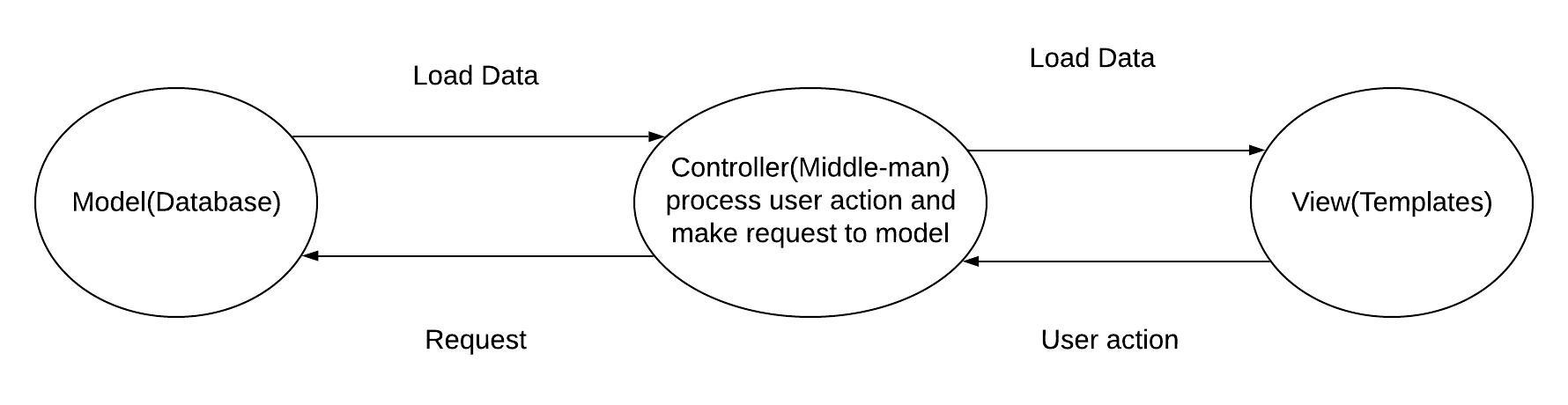
**System Documentation by Lip Chuan Sui**

This section will talk about the MVC framework, database design and overview of final website.

1. **MVC architecture**

MVC stands for Model, View and Controller. MVC architecture separates application into three components, which are Model, View and Controller. Model represents the flow of data. Model store queries for fetching data from database. The data fetched by queries will be sent to Controller. Just think that if you want to fetch or update data in database, you have to write it in Model. The Controller handles the user request and fetches data from Model to load it into View. Just imagine Controller is the main coding part of the system and will handle request from user. Lastly, View is the user interface which contains the templates. View displays data to user and enables user to modifies them. Html pages in View are called templates because View could display different data with same template design for each request. For example, user could create new article post in Yokyo website. Each article page will use same data but display different context. Below is a figure illustrating the interaction between Model, View and Controller. Just imagine Controller is the middle man between Model and View.

Summary of MVC framework



I choose MVC architecture because the architecture distinguishes coding into different parts. For example, if I want to make correction in query, I could straight correct it in Model. If I want to design my templates, I could just do it in View. It is easier for me for coding because the coding is separated base don their natures and function instead of putting everything in one page. Furthermore, CodeIgniter Model uses Active Record when writing queries. Instead of writing full sentence of queries, Active Record queries are written with minimal scripting. This could prevent SQL Injection. Furthermore, CodeIgniter prepares built-in function such as generate random string function, hash password function, sending email function and others.

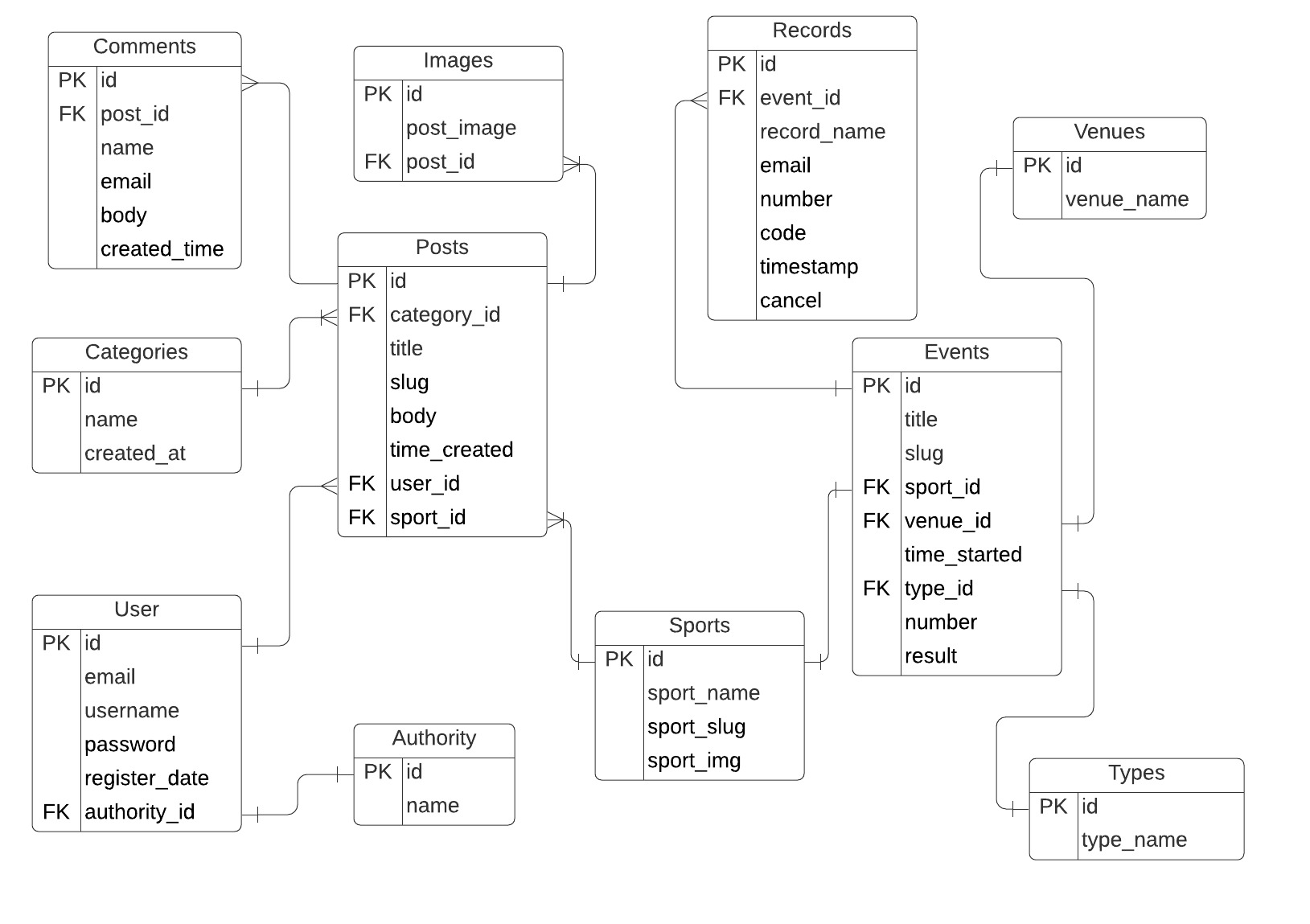
Active record pattern



1. **Database design**

When designing database structure, I want to make sure that each table is fully normalised to BCNF, by specifying all functional dependencies on your tables, with a statement as to whether each table meets BCNF and correctly resolving any normalisation issues.

ER Diagram of Yokyo Database



Example of functional dependecy

Post id → (category\_id, title, slug, body, time\_created, user\_id, sport\_id)

The primary key of post table is Post id. The table is in BCNF because the source of each functional dependency is primary key, which is Post id.

There are some small tables such as types, venues, authority and category. They are designed to be separated tables because the database was reviewed and tested with normalization. Duplicated data was prevented. Besides, user could fetch all venues from venues table to populate option list when choosing venues. Hence, the option list is dynamic instead of hard-coded.

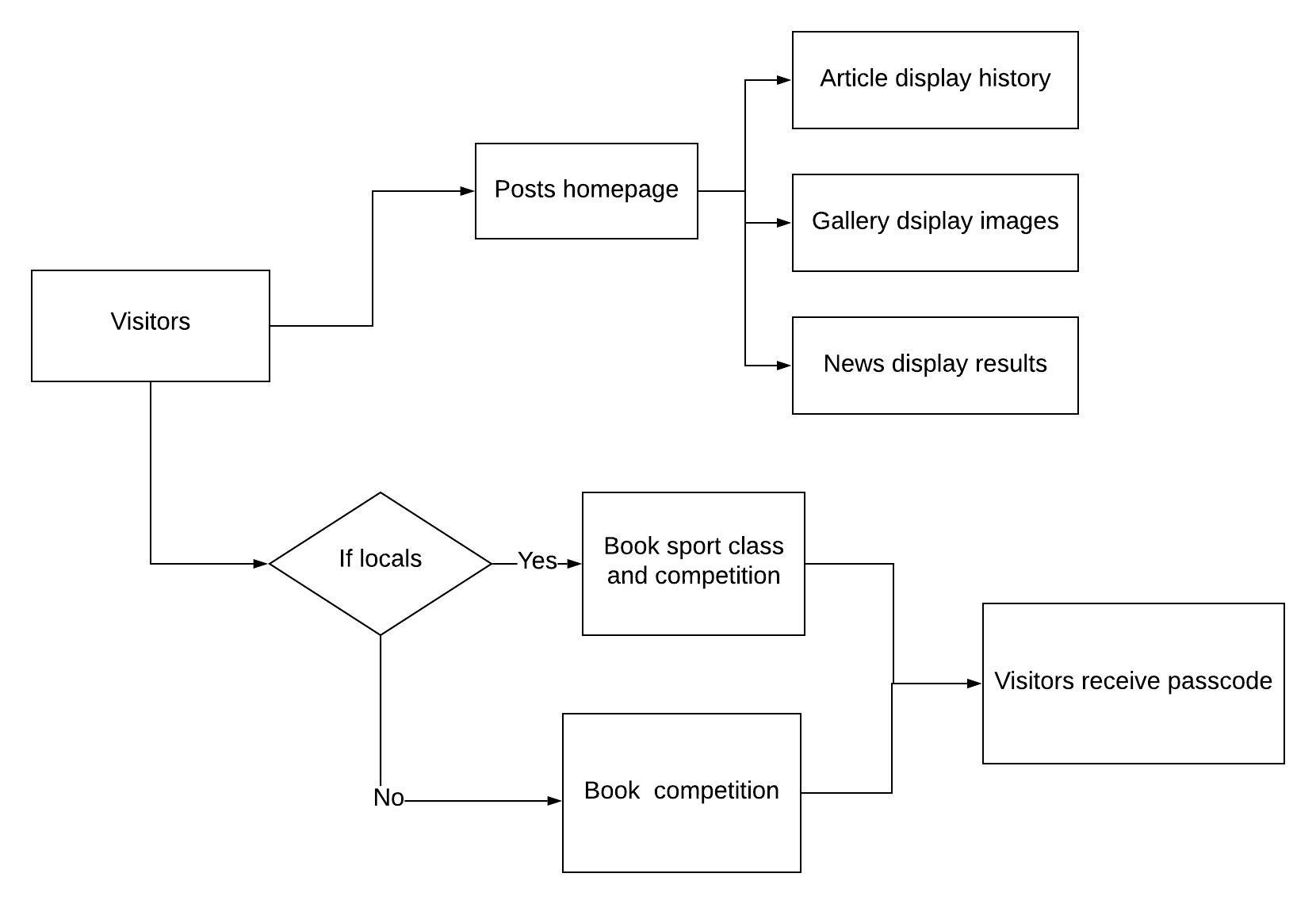
Every relationship between tables are stated clearly. For example, each post could only have 1 category, 1 user, 1 sport but could have multiple comments and images. An event could only have 1 sport, 1 type and 1 venue but could have multiple booking records.

1. **Overview of final website and business process model**

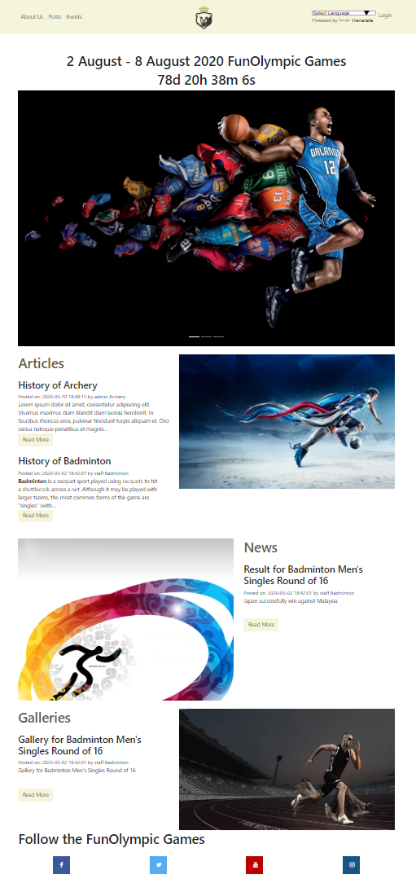
This section will explain how visitors will interact with Yokyo website.

Below is the business model of Yokyo. Visitors could visit posts homepage to look at 3 different categories of posts which are article, gallery and news. Furthermore, visitors could book seat for attending class or competition. Only locals are allowed to book sport class.

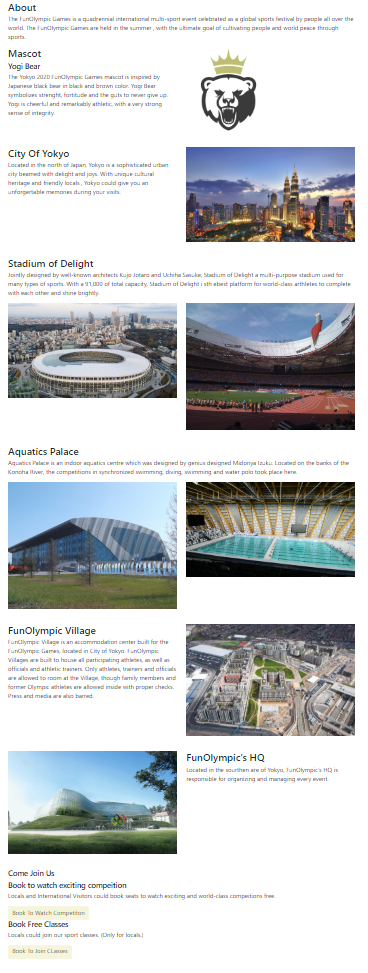
Business model of Yokyo

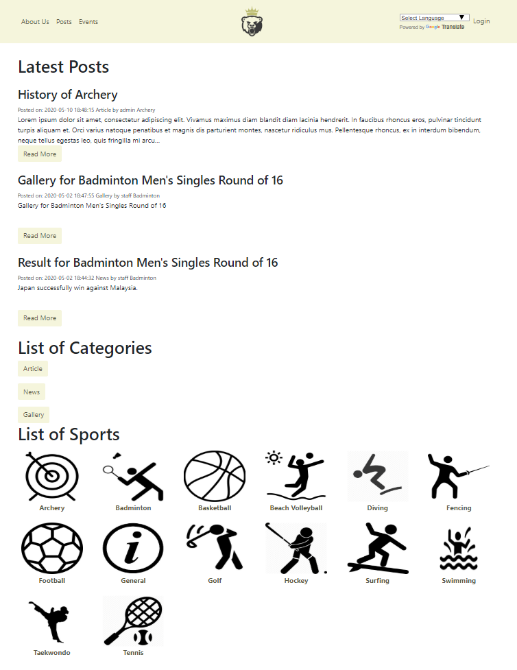


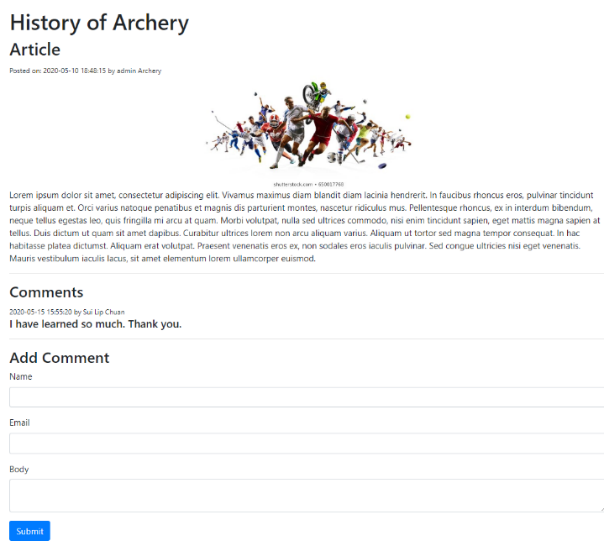
**Visitor interface**



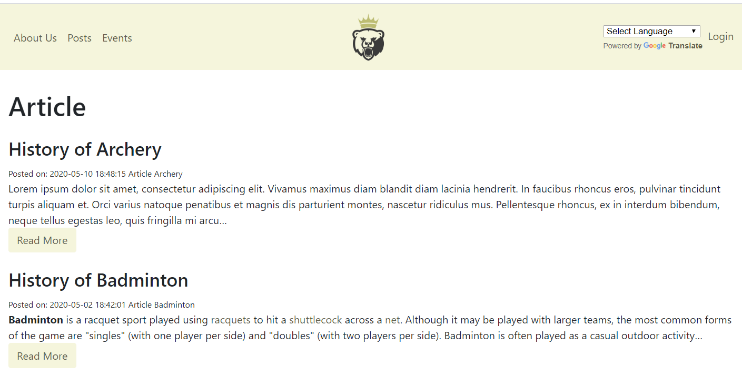
This is the homepage of Yokyo. At the top there is a countdown timer fo the event to start. Middle parts display the newest posts created for each categories. Bottom part are social media button for sharing.

This is the about us page. This page briefly explains the City of Yokyo, the location of events and the mascot. Bottom part display rules about booking. Only locals are allowed to book sport class.

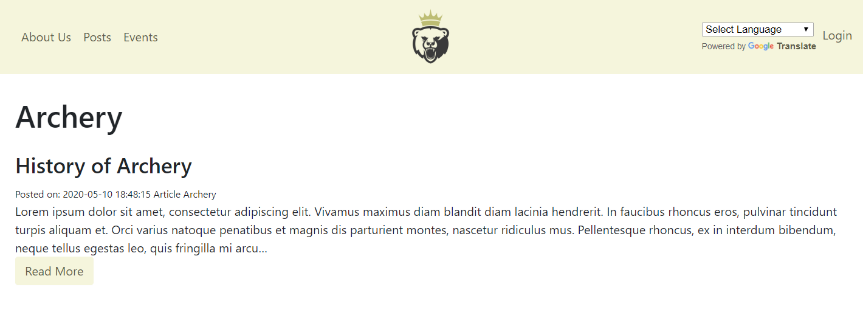
This is the posts homepage of Yokyo. The page display 3 latest posts and contains link to all three categories. Visitors could also choose posts based on sports type at bottom part.



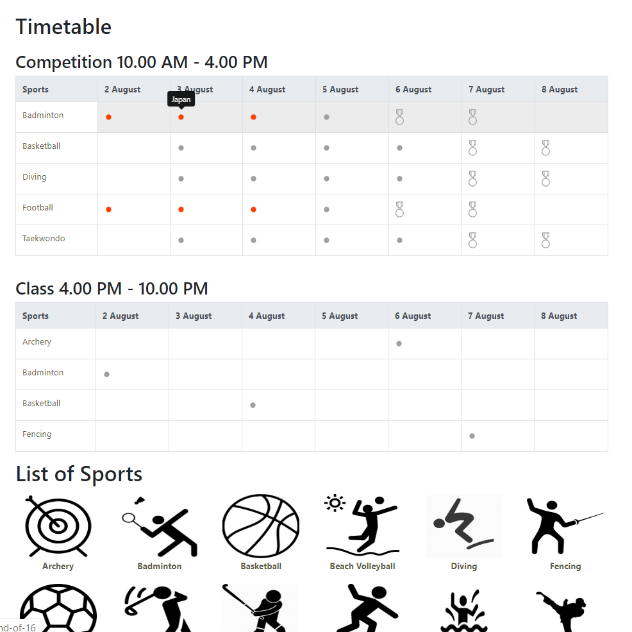
This is an article post. The post display date, title, author, image, body and comment section.



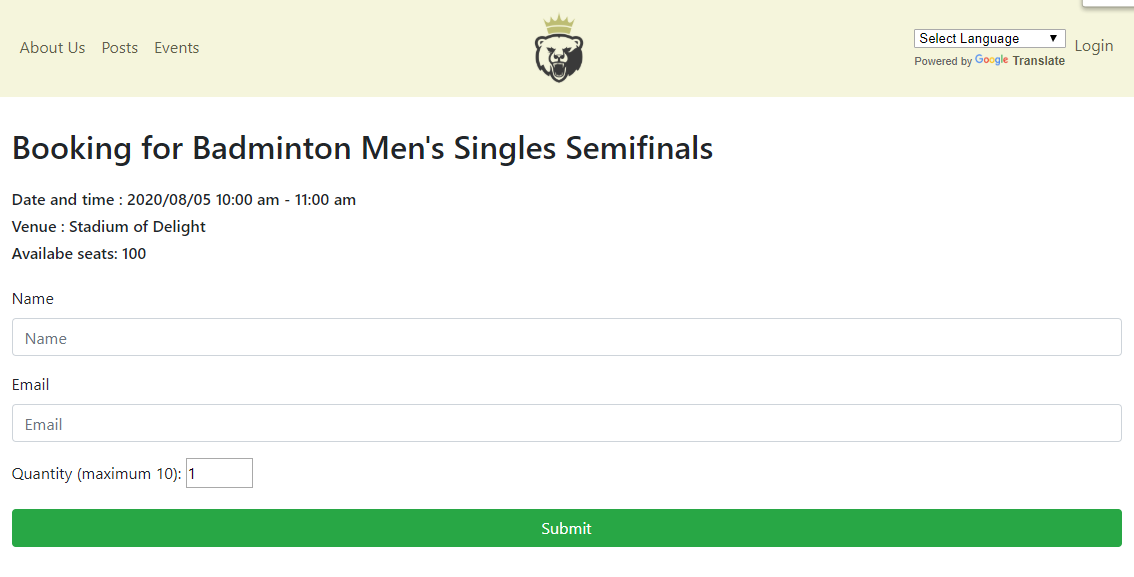
This is the index page for article. The page displays all articles.

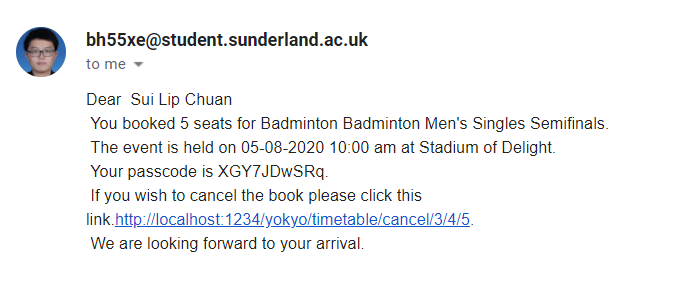


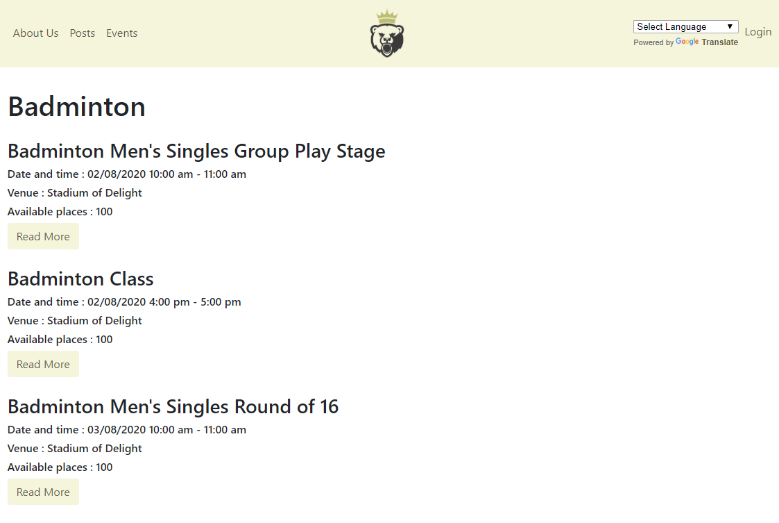
This is the index page for Archery. The page displays all post about the sports.



This is the homepage for events. There two timetables display information about competition and class respectively. Visitors could click on the timetable to book. Red icon means a winner is decided and the match is over. Medal icon means it is a medal-deciding match. Visitor could hover to red icon to know about the winner. Visitor could click on sports on timetable to navigate to page contains only events of that sport.

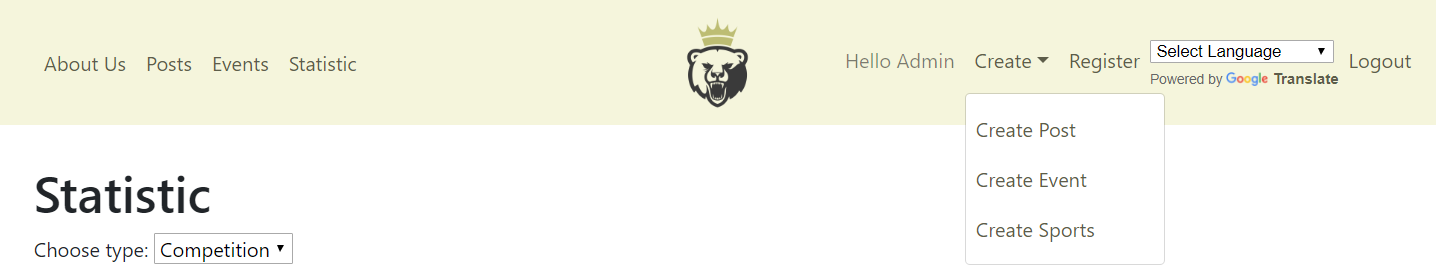
Visitor could book seats in this page. Book button will disable if non-locals intend to book sport class. Visitor could only book a maximum of 10 seats at a time.

After booking seats, visitors will receive email containing passcode. They could choose eot cancel it by clicking on the link.

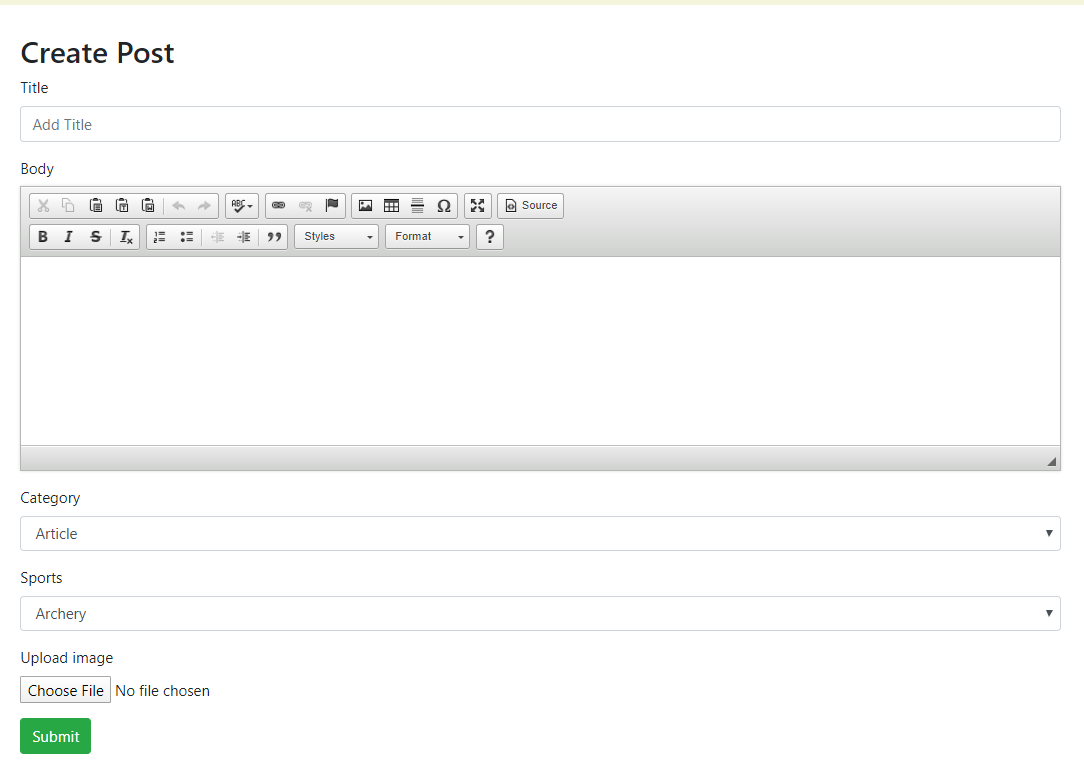
This page displays all events of a sport.

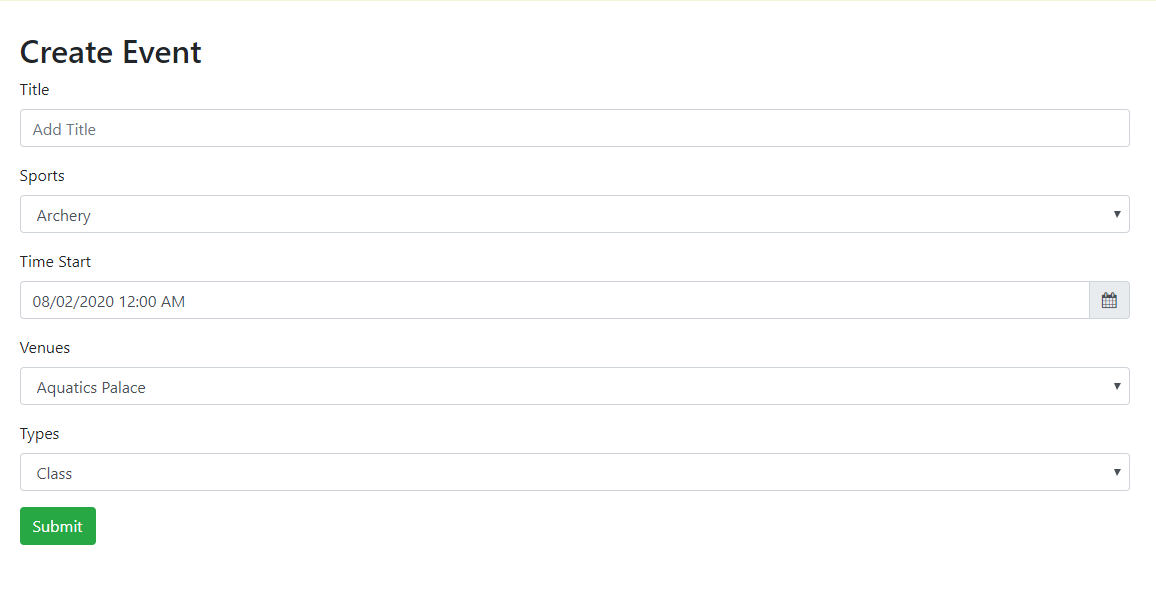
**Admin and staff interface**

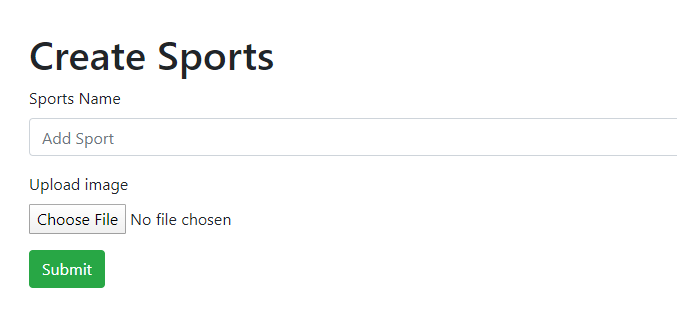
After logging in, the navigation bar will appear differently for admin or staff.



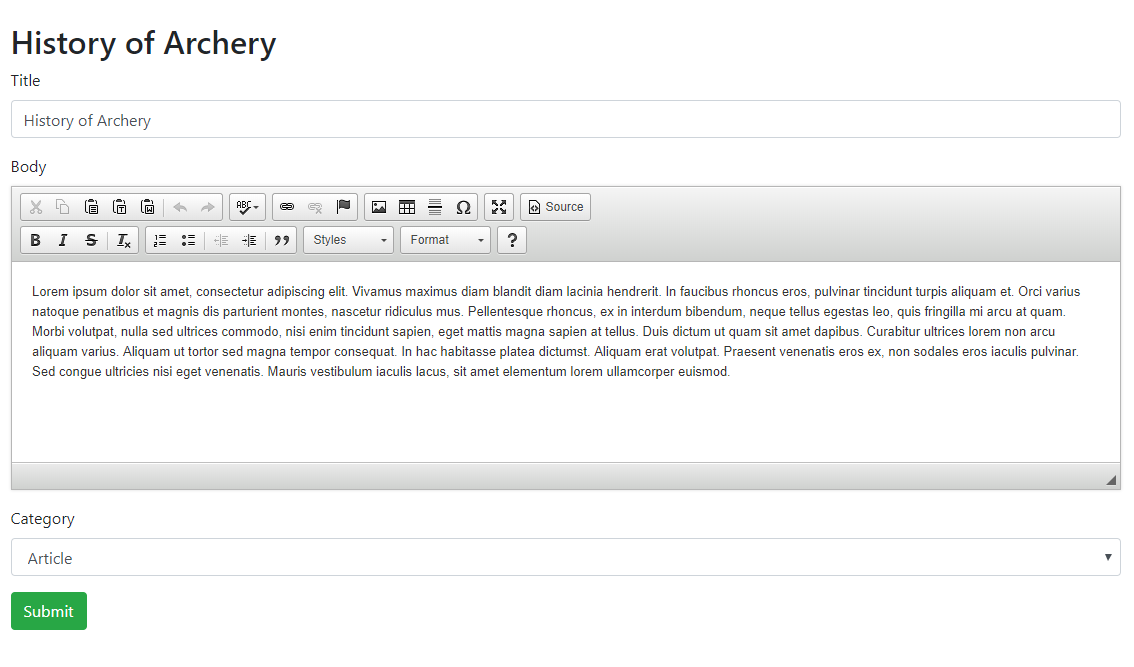
Admin and staff could create posts, sports and events.

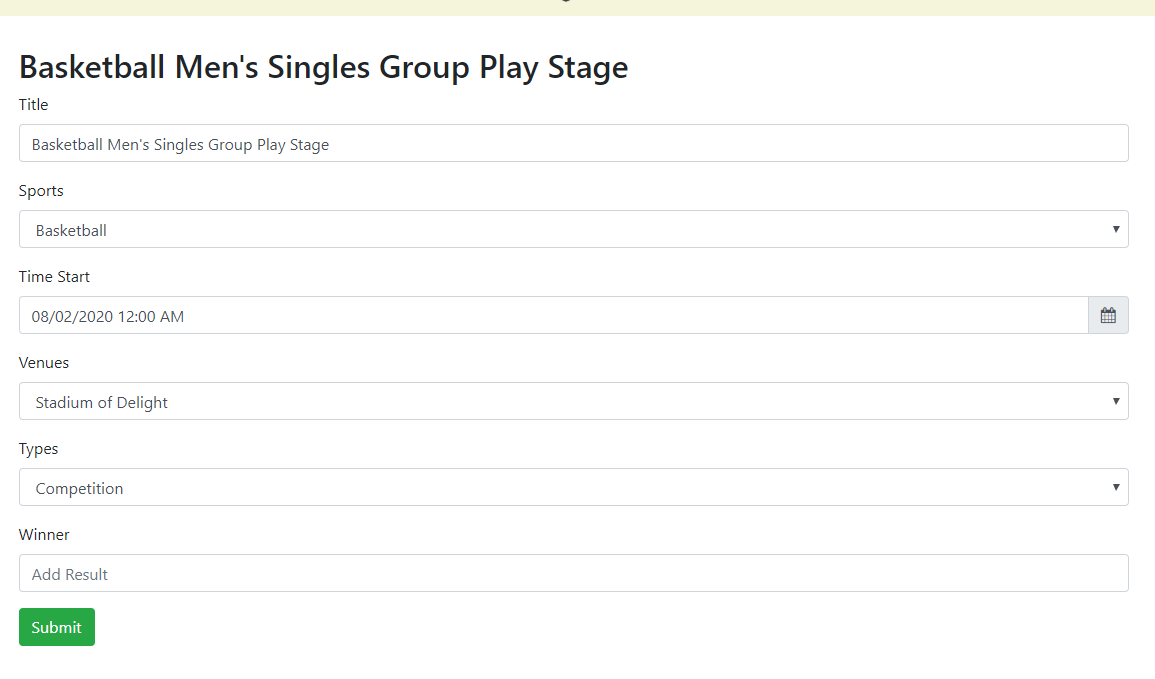






Furthermore, they could edit post and events.





Finally, they could view booking statistic.

