550 Project Milestone 2

Songyang Du, Shengyin Si, Yahan Xu, Yu Yang

March 13, 2023

1. Motivation / description

   Our goal was to provide a simple and user-friendly solution for movie enthusiasts who struggle to find relevant and accurate information online. Our website allows users to search for movies using keywords and displays relevant information, such as poster image, title, ratings, genre, actors, and more.

   Our website is a valuable tool for movie enthusiasts of all levels, offering basic search functionality, daily recommendations, and award records (Oscar). Our additional features enhance the user experience and help users discover new movies they might not have found otherwise.

   Overall, our motivation for this project was to make it easier for people to find and learn about movies, and to create a website that is both informative and enjoyable to use.

2. List of features (for sure)

   - Homepage:
     (a) Daily recommendation: one randomly generated Oscar-winning movie
     (b) List the top 10 recent movies for selected genre
     (c) List the top 10 rated movies for selected language
   - Movies main page:
     (a) Display random generated movie poster
     (b) Customer filter/searching function: title, year, genre etc
   - Movies details page:
     (a) Display the movie's basic information, poster, brief description and related people
   - People's main page
     (a) List of posters for every actor and actresses
     (b) Allows users to search people by name
   - People's detail page
     (a) Display people's basic information, poster, brief description, and related movies
     (b) By clicking the movie poster, the web page will be directed to the it's details page
   - Oscar main page
     (a) Shuffle slides: display some photos, links to other sub-page
     (b) Allows users to search for movies based on different criteria, display posters of the resulting movies
     (c) By clicking the movie poster, the web page will be directed to the it's details page
     (d) Links to other sub pages: a brief history, facts

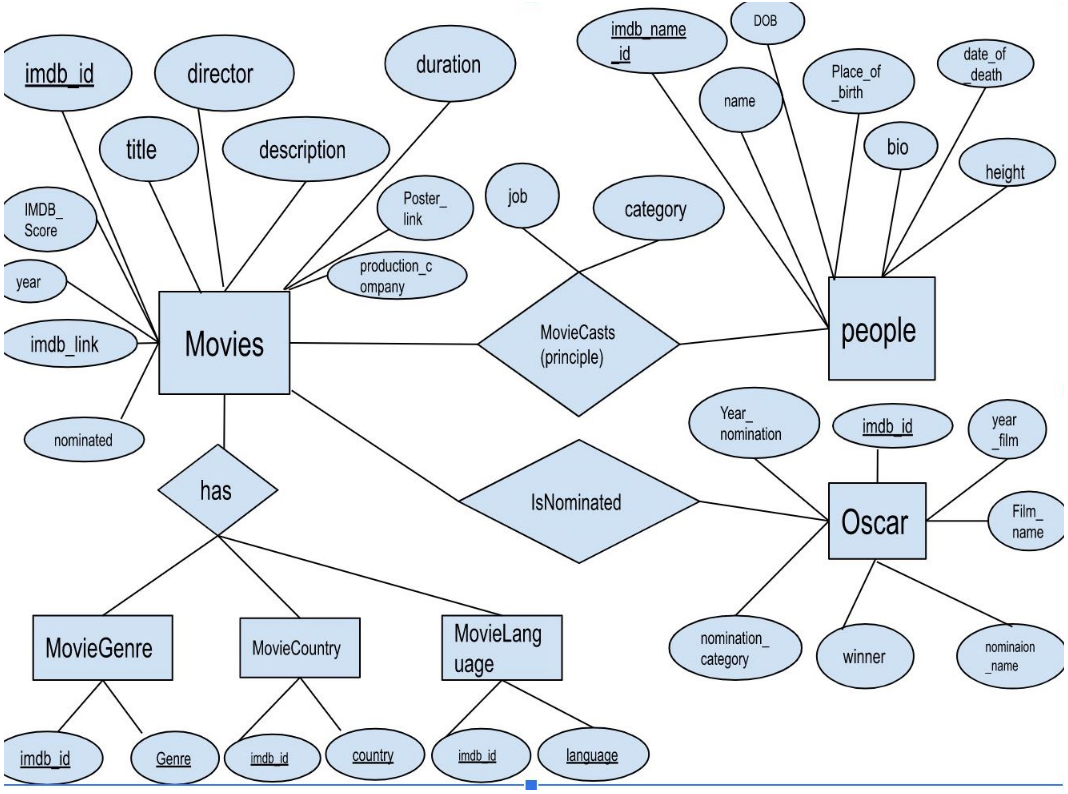
3. List of features (if enough time)

   - Homepage: display the name, age, and profile of people who were born today
   - Movie details page: additional movie recommendation based on the selected movie
   - People: additional information: if participated in any Academy reward movies, number of nominations, number of winning
   - Oscar: additional descriptive analysis of the data

4. List of pages

   - Homepage: click to see the homepage's template
   - Movies: click to see the Movie page's template
   - Movie details page: click to see the movie details page's template
   - People: click to see the people page's template
   - People detail page: click to see the people detail page's template
   - Oscar page: click to see the Oscar page's template

5. Relational schema as ER



6. SQL DDL for creating the DB

```
CREATE TABLE Movies (
Imdb_id            VARCHAR(20)
Title              VARCHAR(250)
Director           VARCHAR(250)
Production_company VARCHAR(250)
Poster Link        VARCHAR(250)
Duration           INT
Imdb_score         INT
Oscar_nominated  BOOLEAN
PRIMARY KEY Imdb_id)

CREATE TABLE MovieGenre (
Imdb_id        VARCHAR(20)
Genre          VARCHAR(250)
PRIMARY KEY Imdb_id, Genre
FOREIGN KEY Imdb_id REFERENCES Movies(Imdb_id))

CREATE TABLE MovieCountry (
Imdb_id        VARCHAR(20)
Country        VARCHAR(250)
PRIMARY KEY Imdb_id, Country
FOREIGN KEY Imdb_id REFERENCES Movies(Imdb_id))

CREATE TABLE MovieLanguage (
Imdb_id     VARCHAR(20)
Language    VARCHAR(250)
PRIMARY KEY Imdb_id, Language
FOREIGN KEY Imdb_id REFERENCES Movies(Imdb_id))
CREATE TABLE MovieCasts (
imdb_id  INT
imdb_name_id  VARVHAR(250)
Category VARVHAR(250)
Job VARVHAR(250)
PRIMARY KEY Imdb_id, imdb_name_id, Category
FOREIGN KEY Imdb_id REFERENCES Movies(Imdb_id)
FOREIGN KEY Imdb_name_id REFERENCES People(Imdb_name_id))

CREATE TABLE People (
Imdb_name_id       INT
Name               VARCHAR(250)
DoB                VARCHAR(250)
Date_of_death      VARCHAR(250)
Place_of_brith     VARCHAR(250)
Height             VARCHAR(250)
Bio                VARCHAR(8000)
Primary Key Imdb_name_id))

CREATE TABLE Oscar (
Imdb_id          INT
Nonimation_category  VARCHAR(250)
Nonimation_name      VARCHAR(250)
Film_name            VARCHAR(250)
Winner               BOOLEAN
screened_year        INT
Ceremony_year        INT
Primary_key Imdb_id
FOREIGN KEY Imdb_id REFERENCES Movies(Imdb_id))
```

7. Data Cleansing

   - Multiple datasets contain information about movies, we will merge those data sets by some common columns(primary key, eg:imdb_id), and remove duplicates.
   - Remove duplicates and redundancy in the Peoeple dataset.
   - No imdb_id in the Oscar database:
     (a) Use year and movie name as the merging key to attach the Imdb_id from the Movie table.
     (b) Remove any invalid data.
     (c) Create a boolean attribute in the Movie table to indicate whether the movie has any kind of Oscar nomination or not.
   - Using Python to scrape the movie descriptions from IMDb.
   - Using Python to scrape the URL of people's profile photos from IMDb, and create an additional column Photo in the People table to store the URL.

8. List of technologies

   - Python: data cleansing and data crawling
   - MySql: managing the data
   - Node.js: running environment for JavaScript
   - React: building interactive user interface
   - Css: styling and formatting web pages
   - Html: creating structures and contents of the webpage
   - MUL: building visually appealing web applications
   - Recharts: create interactive and responsive charts and graphs
   - Bootstrap: grid system to create some responsive HTML templates, eg: carousel, shuffle slides, customized displays for forms etc.

9. Description of what each group member will be responsible for

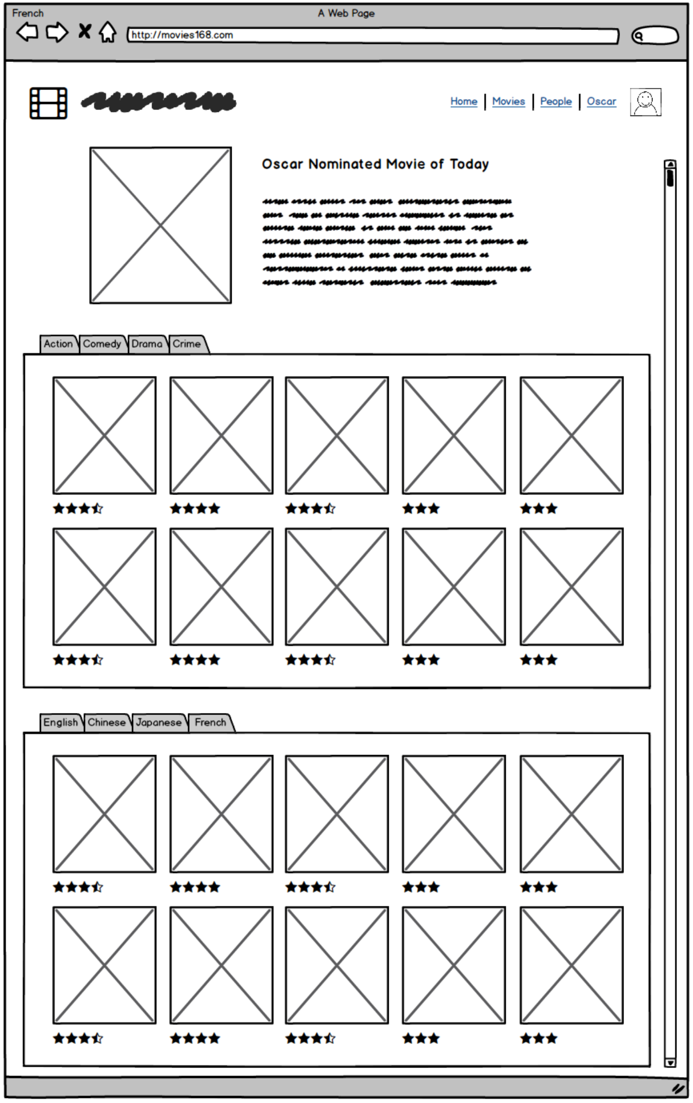
   We need the responsibilities below to complete this web application:

- UI/UX: create an attractive and intuitive interface for the web application
- Database Developer: Use Python for data cleaning and processing. Creates the database schema, and writes optimized SQL queries.
- Back-end: set up server-side using Node.js, manage MySQL database, define API endpoints
- Front-end: build UI using React and Material UI, style with HTML and CSS, implement client-side routing, use Recharts for visualizations
- Quality assurance: Create test plans and cases, conduct testing, identify and fix issues

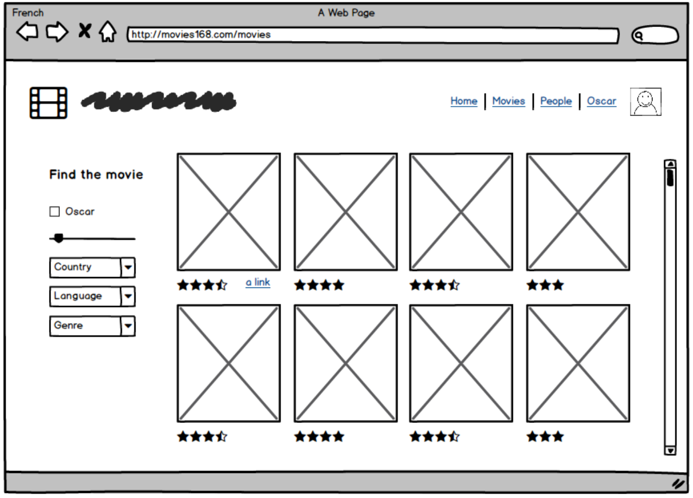
We believe it's a good idea to rotate the roles among the team members throughout the project to encourage learning and collaboration so we decided to assign tasks based on the pages of this web application:

- Homepage: Shengyin Si
- Movies: Songyang Du
- People: Yu Yang
- Oscar: Yahan Xu

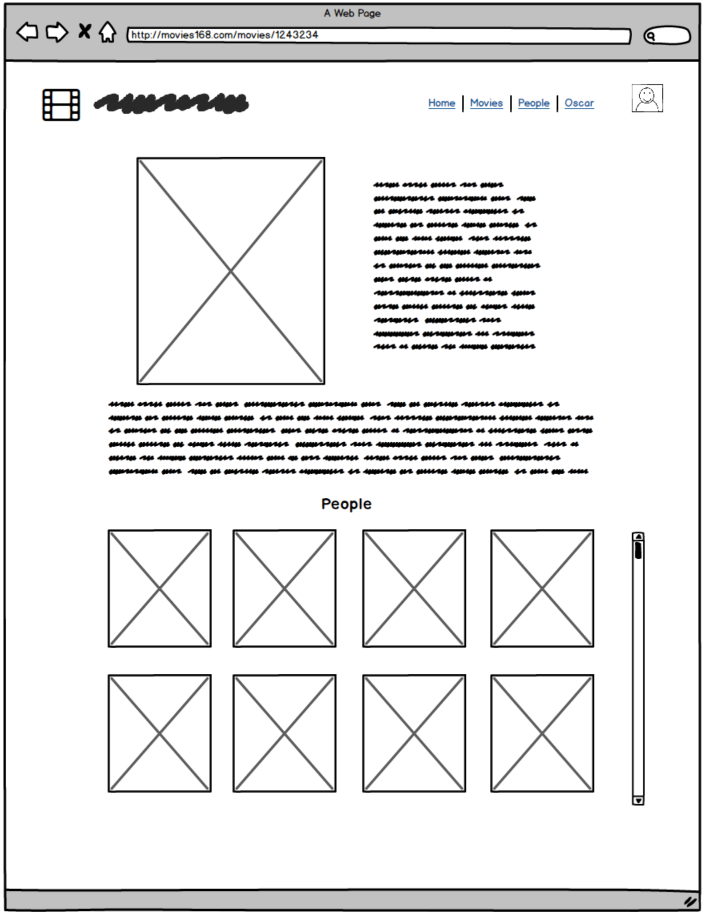
- Homepage template



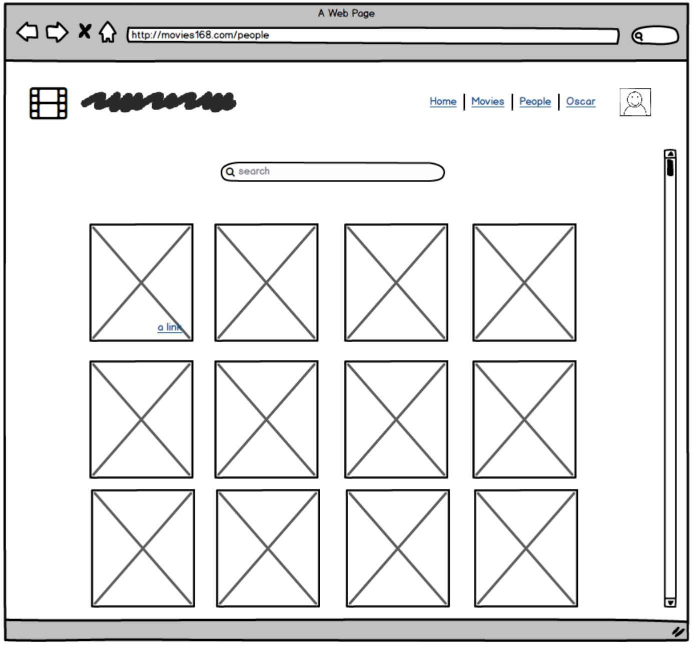
- Movies page template



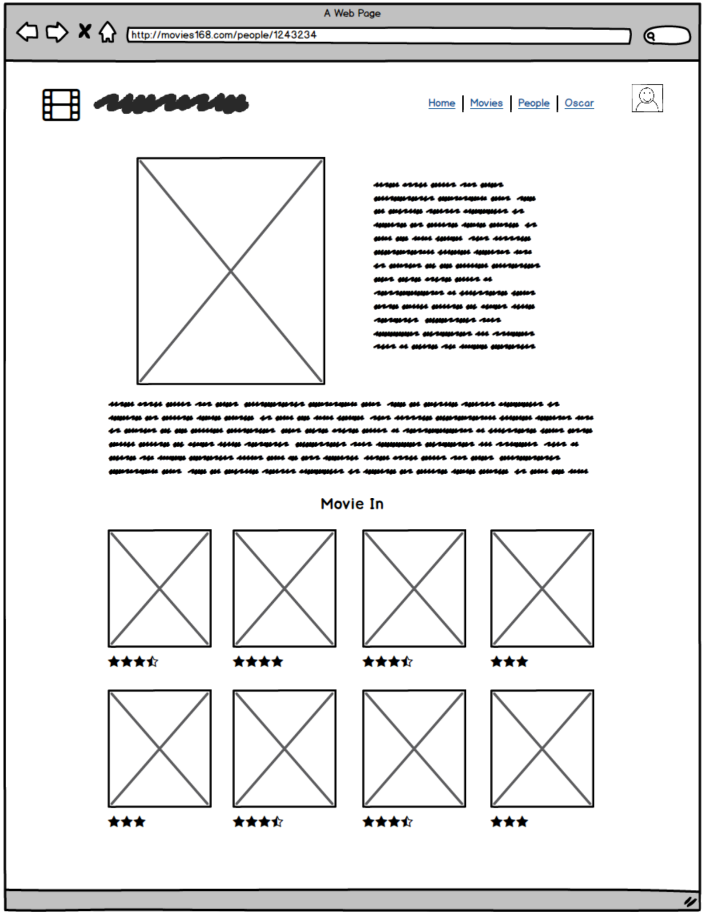
- Movies detail page



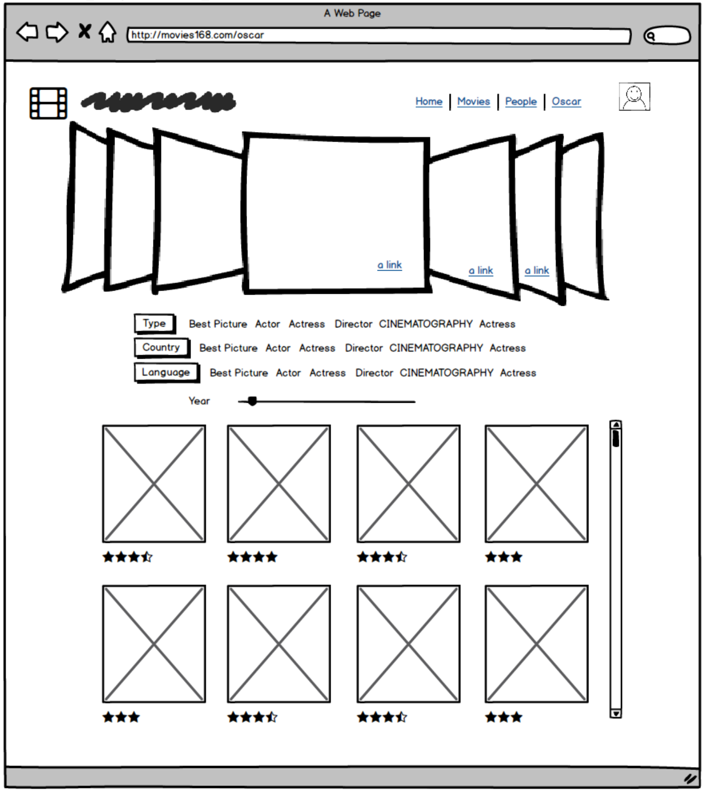
- People page template



- People detail page


A Web Page
http://movies168.com/people/1243234
Home | Movies | People | Oscar
Movie In

- Oscar page template


A Web Page
http://movies168.com/oscar
Home | Movies | People | Oscar
a link
a link
a link
Type Best Picture Actor Actress Director CINEMATOGRAPHY Actress
Country Best Picture Actor Actress Director CINEMATOGRAPHY Actress
Language Best Picture Actor Actress Director CINEMATOGRAPHY Actress
Year