



Project Implementation Report

IST 659 M002

Milan Fashion Week

By

The Lightbringers

Ishita Joshi, Rutwik Ghag, Siddhi Bhandari

Table of Contents

1. Project Summary
2. Conceptual Data Model Diagram
3. Tools Used
4. Data Dictionary
5. Business Rules
6. SQL Code
7. Application Screens
8. Team Log

Project Summary

Milan Fashion Week is a showcase for the world's top designers which also offers a chance for emerging designers to grab attention. Each year it offers more, with live events and shopping opportunities and live streaming, so attendees can celebrate first hand Fashion Week and Milan as the global capital of creativity. It's estimated around 5000 visitors attend each year. All this requires a great deal of event planning. Behind all the glitz and catwalks you'll find a highly polished team of event management professionals who have to plan all tasks needed to make an event happen. They will cover everything from conception through to completion, such as site planning, catering, health & safety, creating schedules, managing contracts with suppliers and making sure the VIP guests receive VIP treatment! All these can be stressful if not organized properly. It is extremely important to keep all the brands, models and agencies satisfied. A tiny misplace of a tag can lead to a bigger problem.

That's where we step in, the Lightbringers. Being interested in fashion and having a sense of understanding for the underlying issues, we decided to come up with a database management system for the Milan Fashion Week and similar events. The aim of this project is to ensure computerization of the backend at the fashion week. The project acts as a guide, not only for the organizers but also for the visitors. The module can be used for handling any fashion events. We provide solutions not only for the management of the event, but also for deciding winners at the event. It is a software that provides overall support needed for a successful fashion event. It may also be used casually by visitors to gather details about certain looks, brands, and other details based on season, gender, and sizes, etc.

The Database includes the following tables:

- Models: Contains data about the model like model id, name, gender, the size that they wear, payment, and the brands they are associated with
- Managers: Manager id, models they manage, their fees, and the agency they are associated with
- Brands: the type of clothes they will display, the item sizes, and the season
- Designers: their designing forte, the brand they are associated with, staff working with them, as well as their email
- Judges: the season they are judging, the model they voted for, email, staff allotted to them
- Staff: who they are allocated to, and their department
- Guests: their status, seating priority, staff allotted to them, their vote

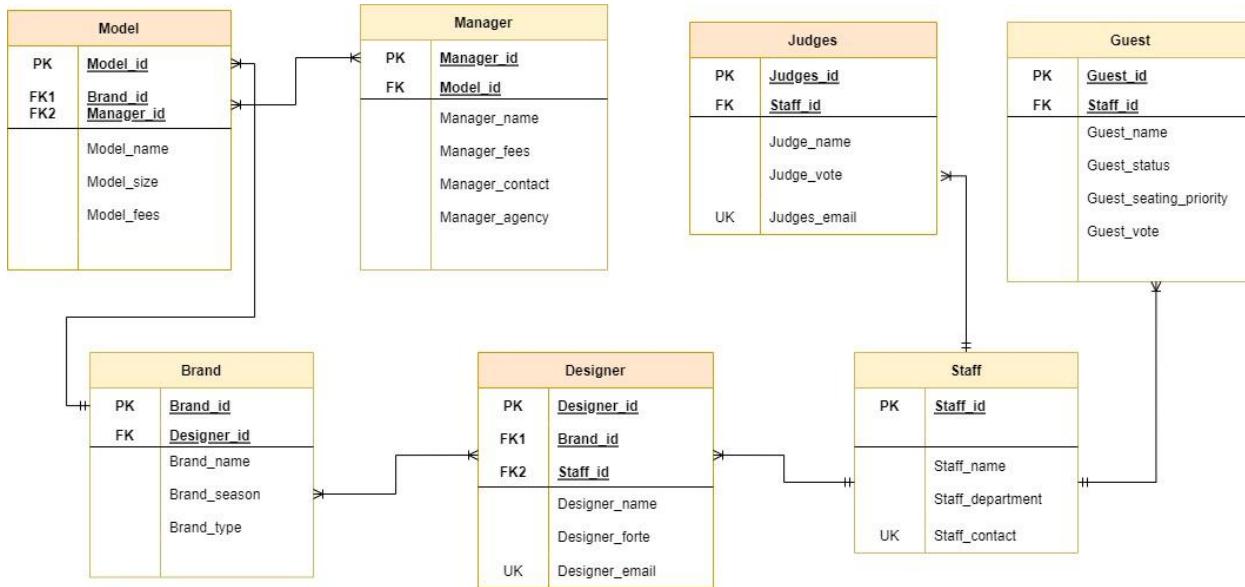
Deliverable:

On-demand data whatever is required

Winners calculated by votes

Pay calculations

Conceptual Data Model:



The data model provided us with a blueprint for the finished database, ensuring consistency, and making sure that no elements have been forgotten.

All the tables have the id as their primary key.

Tools Used:

- Azure Data Studio: SQL queries
- Draw.io: Conceptual and logical diagrams
- Webflow: User Interface
- Microsoft Teams, Google docs, Slides: Collaboration, Report, Presentation

Data Dictionary

Table: Models

Field Name	Data type	Field length	Constraints	Description
Model_id	Int		Not null Primary key	Identification number for the models
Model_name	Varchar	50	Not null	Name of model

Manager_id	Int		Not null	ID of the assigned manager
Gender	Varchar	50	Not null	Gender
Model_size	Varchar	50	Not null	Clothing size
Brand_id	Int		Not null	ID for the brand worn
Model_fees	Int		Not null Greater than zero	Fees charged in USD

Table: Manager

Field Name	Data type	Field length	Constraints	Description
Manager_id	Int		Not null Primary key	Identification number for the manager
Manager_name	Varchar	50	Not null	Name of the manager
Manager_fees	Int		Not null Greater than zero	Fees charged in USD
Manager_contact	Varchar	50	Not null	Email address for point of contact

			Unique	
Manager_agency	Varchar	50	Not null	The agency they are associated with

Table: Judges

Field Name	Data type	Field length	Constraints	Description
Judge_id	Int		Not null Primary key	Identification number for the judge
Judge_name	Varchar	50	Not null	Name of the judge
Staff_id	Int		Not null Foreign key	ID of the staff allocated
Vote	Int		Not null	Vote given to the model
Judge_contact	Varchar	50	Not null Unique	Contact information

Table: Guests

Field Name	Data type	Field length	Constraints	Description
Guest_id	Int		Not null	Identification number for the guest
Guest_name	Varchar	50	Not null	Name of the guest
Seating_priority	Int		Not null	Priority for the services provided and seating booth
Staff_id	Int		Not null	ID of the staff allocated
Poll	Int		Not null	Vote given to the model

Table: Brand

Field Name	Data type	Field length	Constraints	Description
Brand_id	Int		Not null	ID for the brand worn
Brand_name	Varchar	50	Not null	Name of the brand
Designer_id	Int		Not null	Identification of the designer

Brand_season	Varchar	50	Not null	Seasonal representation of the brand
Brand_forte	Varchar	50	Not null	The speciality of the brand

Table: Designer

Field Name	Data type	Field length	Constraints	Description
Designer_id	Int		Not null	Identification of the designer
Designer_name	Varchar	50	Not null	Name of the designer
Brand_id	Int		Not null	ID for the brand worn
Staff_id	Int		Not null	ID of the staff allocated
Designer_forte	Varchar	50	Not null	The speciality of the designer
Designer_contact	Varchar	50	Not null	Contact information of the designer

Table: Staff

Field Name	Data type	Field length	Constraints	Description
Staff_id	Int		Not null	ID of the staff allocated
Staff_name	varchar	50	Not Null	Name of the staff
Department	varchar	50	Not null	The department they are allotted
Staff_contact	varchar	50	Not null	Staff's contact information

Business Rules:

- Model and manager salaries cannot be negative.
- No repeated entries can be added.
- ID is necessary for adding, updating or deleting an entry.
- Winners are calculated based on votes by judges and audience.
- One or more staff members are allocated to multiple designers, judges as well as the guests.
- The judges cannot communicate directly with the designers or the models in order to maintain legitimacy.

SQL Code:

1. To find models paid more than \$15,000

```
select modelname as Model, gender as Gender
from Models
where modelfees > 15000
group by gender, modelname
```

2. To find the average pay by gender for models

```
select avg(modelfees) as Pay, gender as Gender
from Models
group by gender
```

3. To find the average pay by season for models.

```
select avg(m.modelfees) as Pay, b.brandseason as Season  
from Models m join Brands b on m.brandid = b.brandid  
group by b.brandseason
```

4. To find the brand each model will represent and the season

```
select Models.modelname as Model, Brands.brandname as Brand, Brands.brandseason as Fashion_Season  
from Models JOIN Brands on Models.brandid=Brands.brandid  
order by brandname
```

5. To find the most voted model

```
with model_query AS(  
select vote,modelname  
from Judges j JOIN Models m on j.vote = m.modelid  
)  
SELECT count(vote) OVER(partition by modelname)AS count_of_votes,modelname,vote  
FROM model_query  
order by count_of_votes DESC
```

Drop view if EXISTS something

GO

Create view something AS

```
with model_query AS(  
select vote,modelname  
from Judges j JOIN Models m on j.vote = m.modelid )  
SELECT count(vote) OVER(partition by modelname)AS count_of_votes,modelname,vote  
FROM model_query  
GO
```

```
Select distinct count_of_votes as No_of_votes, modelname as Model from something  
order by count_of_votes DESC
```

6. To find the winning designer

```
select distinct d.designername as Designer, count_of_votes as Votes  
from something s join Models m on s.vote = m.modelid join Brands b on m.brandid = b.brandid join Designers d on  
b.designerid = d.designerid  
order by count_of_votes desc
```

Screen Layouts

TLB

Home Explore

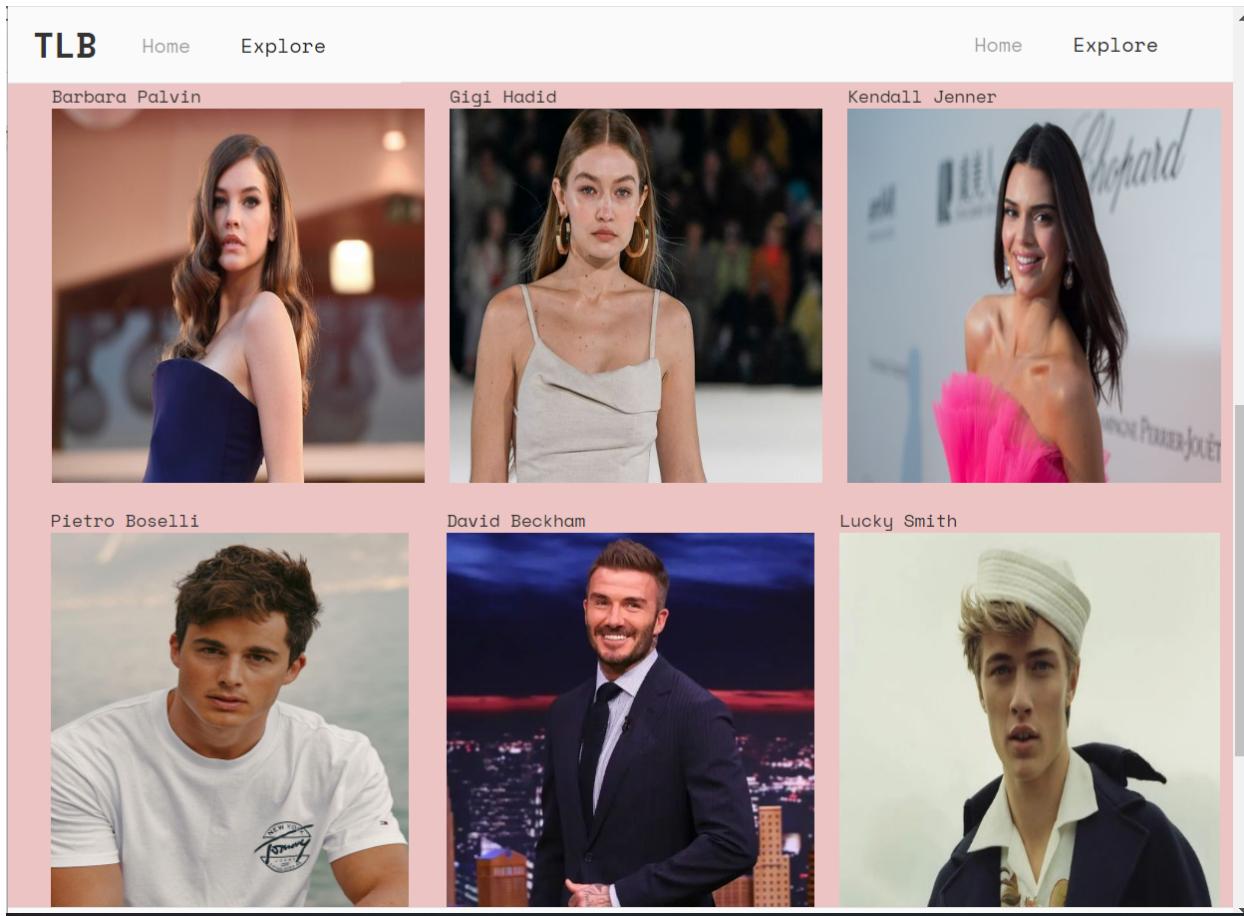
THE LIGHTBRINGERS
PRESENT

Milan Fashion Week

Explore



Landing page



Home screen with models from the database

Subscribe

First Name

Last Name

E-Mail

Subscribe

TLB

[Privacy Policy](#)

[Terms & Conditions](#)

[Bookings](#)

[Events](#)



Site Footer

TLB

Home Explore

Models

Add Delete Update

Name

Model ID

Size

Fees

Submit



This screenshot shows a user interface for managing model data. At the top, there's a header with the logo 'TLB' and navigation links 'Home' and 'Explore'. Below the header, the title 'Models' is displayed. A horizontal bar contains three buttons: 'Add', 'Delete', and 'Update'. The main area has four input fields labeled 'Name', 'Model ID', 'Size', and 'Fees', each with an associated text input box. A 'Submit' button is located at the bottom left of this section. To the right of the input fields is a photograph of a fashion runway show with several models in vibrant, sequined gowns.

Add, delete, and update module for models and other tables

TLB

Home Explore

Gigi Hadid

Wearing: Gucci

< Jacques's Menswear Show Size: S Designer: Judith Leiber >



This screenshot shows a detailed view of a specific model, Gigi Hadid. The page title is 'Gigi Hadid' with the subtitle 'Wearing: Gucci'. Below the title, there are navigation arrows and text indicating the event is 'Jacques's Menswear Show', the size is 'S', and the designer is 'Judith Leiber'. To the right of the text is a large, clear photograph of Gigi Hadid walking down a runway, wearing a light-colored, draped, one-shoulder dress. The background is blurred, showing other people and lights typical of a runway show.

Description of the model and their look

Top Models

Season ▾ Gender ▾ Votes ▾ Pay ▾

Audience

Judges

Filtering models based on season, gender, most voted by audience and judges, pay.

Top Models

Season ▾ Gender ▾ Votes ▾ Pay ▾

Audience

Judges

Most voted model



Result for the most voted model by Judges

Team Log

Task	When	Performed by
Deciding the topic and brief research about the project.	04/02/2022	I, S, R
Creating the conceptual model	04/12/2022	I, S
Creating the logical model and defining the relationship between the entities	04/20/2022	I, S, R
Creating database, the up-down script to create the tables required and data entries	04/25/2022	I, R
Creating primary keys, foreign keys, and other required constraints	04/28/2022	S, R
Formulating the business rules and questions to be answered	05/01/2022	I, S, R
Establishing connection with Azure data studio and execution of the queries	05/05/2022	I, S, R
Designing the user interface	05/10/2022	S
Creating the PowerPoint presentation	05/27/2022	I
Creating the report document	05/27/2022	I, S, R
Final presentation and demo	05/29/2022	I, S, R