

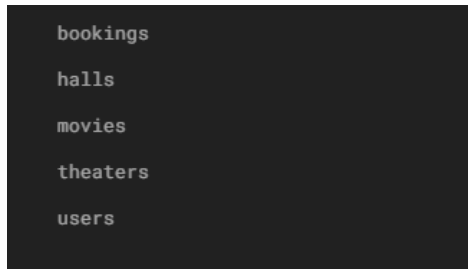
Student id : 65130500054

name : Phuttinan Phaksaweang

Application : Movie Ticket Booking

Description : เป็นระบบจองตั๋วหนัง

Collection :



movies:

Column	Type	Description
title	STRING	ชื่อหนัง
startDate	TIMESTAMP	วันที่เริ่มฉาย
endDate	TIMESTAMP	วันที่จบการฉาย
languages	ARRAY<STRING>	ภาษาที่รองรับ
cast	ARRAY<STRING>	นักแสดง
genres	ARRAY<STRING>	ประเภทหนัง
extract	STRING	เรื่องย่อ
thumbnail	STRING	URL รูปภาพ

theaters

Column	Type	Description
theaterName	STRING	ชื่อโรงหนัง
location	OBJECT	ตำแหน่ง โรงหนัง
movieTitle	ARRAY<STRING>	ชื่อหนังที่ฉายทั้งหมดในโรงหนัง
movies	ARRAY<OBJECT>	รายการหนังที่ฉายและรายละเอียด

halls

Column	Type	Description
hallName	STRING	ชื่อ Hall
projection	STRING	รูปแบบการฉายหนังเช่น IMAX ,2D
seatPrice	NUMBER	ราคาต่อ 1 ที่นั่ง
seatingPlan	OBJECT	รูปแบบที่นั่ง เช่น {rows:10, seatsPerRow:15}
theaterName	STRING	ชื่อ โรงหนังที่ Hall นี้อยู่

movies:

Column	Type	Description
hall	STRING	ชื่อ Hall ที่ต้องเข้าชม
movieTitle	STRING	ชื่อหนังที่ต้องชม
showtime	TIMESTAMP	วันเวลาที่ฉาย
bookedSeats	ARRAY<STRING>	ที่นั่งที่จองทั้งหมด
status	boolean	สถานะของตั๋วว่าใช้หรือยัง
theaterName	STRING	ชื่อโรงหนัง
timestamp	TIMESTAMP	วันเวลาจอง
totalPrice	NUMBER	ราคาที่ต้องจ่าย
userId	STRING	ID เจ้าของคนจอง

users

Column	Type	Description
email	STRING	email
role	STRING	ตำแหน่งในการเข้าถึงข้อมูล

Design Patterns

1. Extended Reference Pattern

```
// ใน theatersData:
{
  theaterName: "Major Cineplex Ratchayothin",
  // เก็บรายชื่อหนังไว้อ้างอิง
  movieTitle: [
    "The Grudge",
    "The Gentlemen",
    //...
  ],
  // เก็บข้อมูลรอบฉายพร้อมข้อมูลหนังบางส่วน
  movies: [
    {
      movieTitle: "The Grudge",
      hall: "Hall A",
      time: ["12:00:00", "15:00:00"],
      language: "English"
    }
    //...
  ]
}
```

- มีการเก็บข้อมูลอ้างอิงของหนัง (movieTitle) ไว้ใน theater document
- ช่วยลดการ join เวลาต้องการดูว่าโรงหนังฉายหนังอะไรบ้าง

2. Subset Pattern

```
// ข้อมูลเต็มใน moviesData:
{
  title: "The Grudge",
  startDate: "2024-12-01",
  endDate: "2024-12-31",
  languages: ["English", "Thai"],
  cast: ["Andrea Riseborough", ...],
  genres: ["Horror", "Supernatural"],
  extract: "...",
  thumbnail: "..."
}

// ข้อมูลย่อยใน theatersData.movies:
{
  movieTitle: "The Grudge",
  hall: "Hall A",
  time: ["12:00:00", "15:00:00"],
  language: "English"
}
```

- เก็บเฉพาะข้อมูลที่จำเป็นสำหรับการแสดงรอบฉาย
- ไม่ต้องเก็บข้อมูลเต็มที่ไม่จำเป็น เช่น cast, extract

3. Outlier Pattern

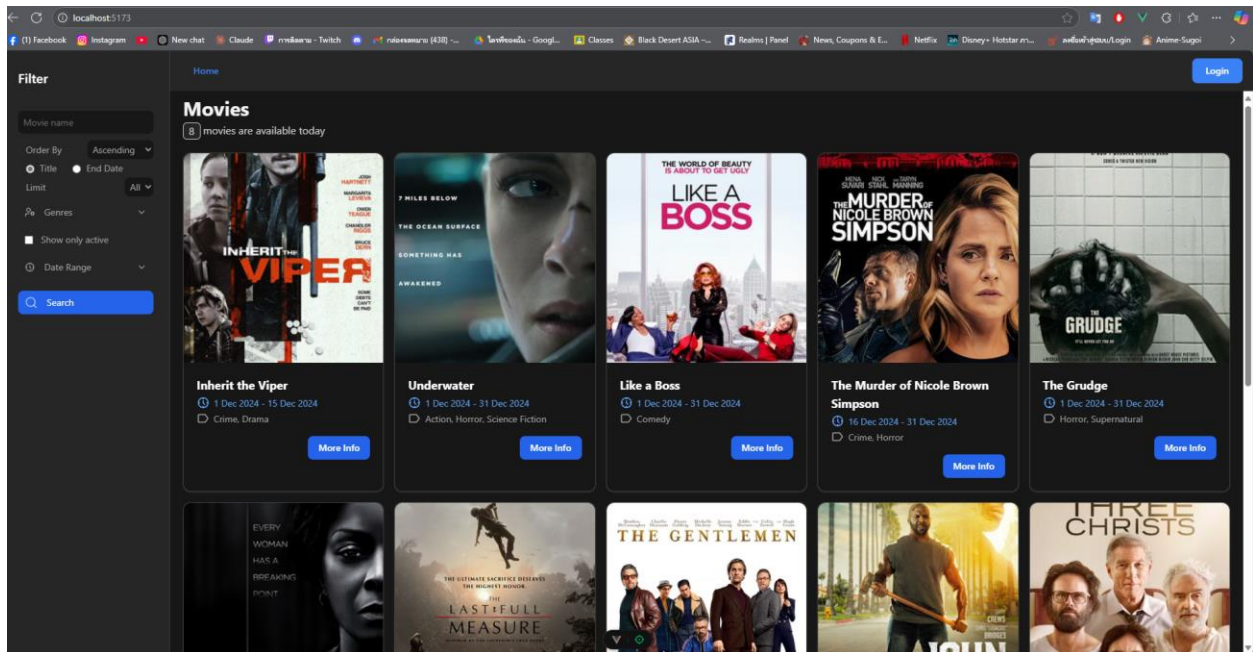
```
// แยกข้อมูล hall ออกมาเป็น hallsData:
{
  theaterName: "Major Cineplex Ratchayothin",
  hallName: "Hall A",
  seatingPlan: { rows: 10, seatsPerRow: 15 },
  projection: "IMAX LAZER 2D",
},
```

- แยกข้อมูลรายละเอียดของ hall ออกมาเก็บต่างหาก
- ลดขนาดของ theaters document หลัก

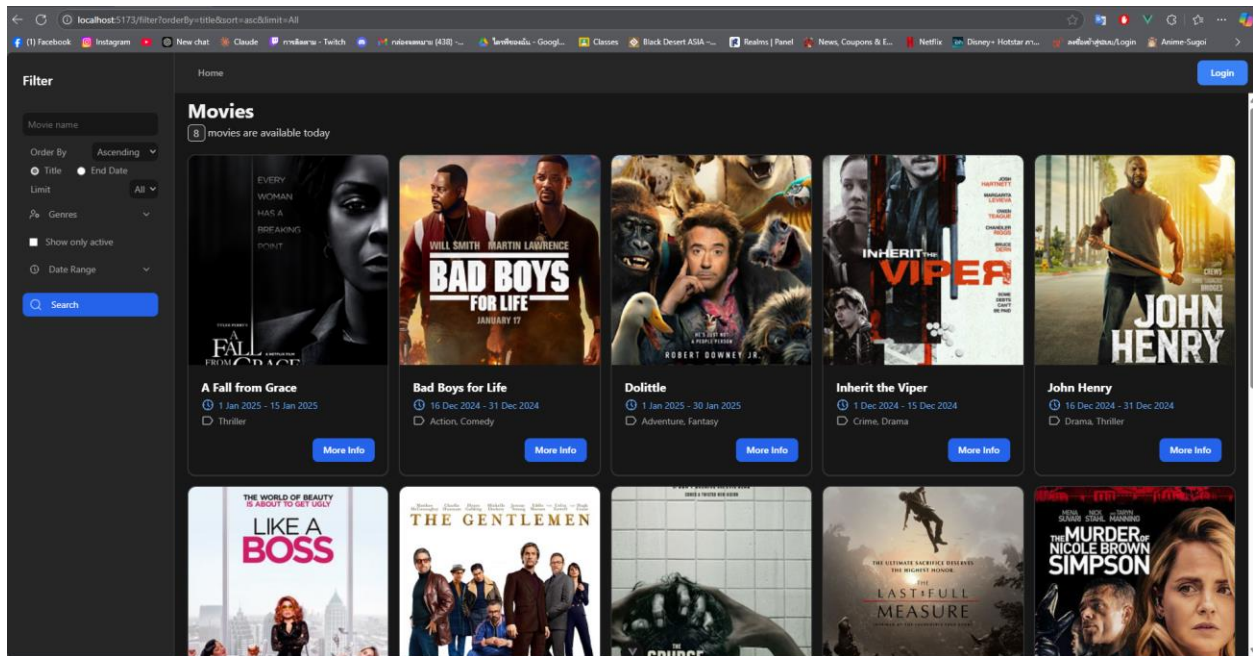
Queries

เนื่องจากระบบนี้การกำหนด Query แบบเจาะจง เพื่อ Filter ไว้มีทางเลือกน้อยไม่ค่อยเหมาะกับ business นี้ เลยจัดทำแบบ custom Query เลือกอะไรจะไปเป็น where ตอน query

Filter ที่ 1 OrderBy เมื่อ กด Search ทำการจัดเรียงตามที่เราเลือกเช่น OrderBy Title , จัดเรียงแบบ asc



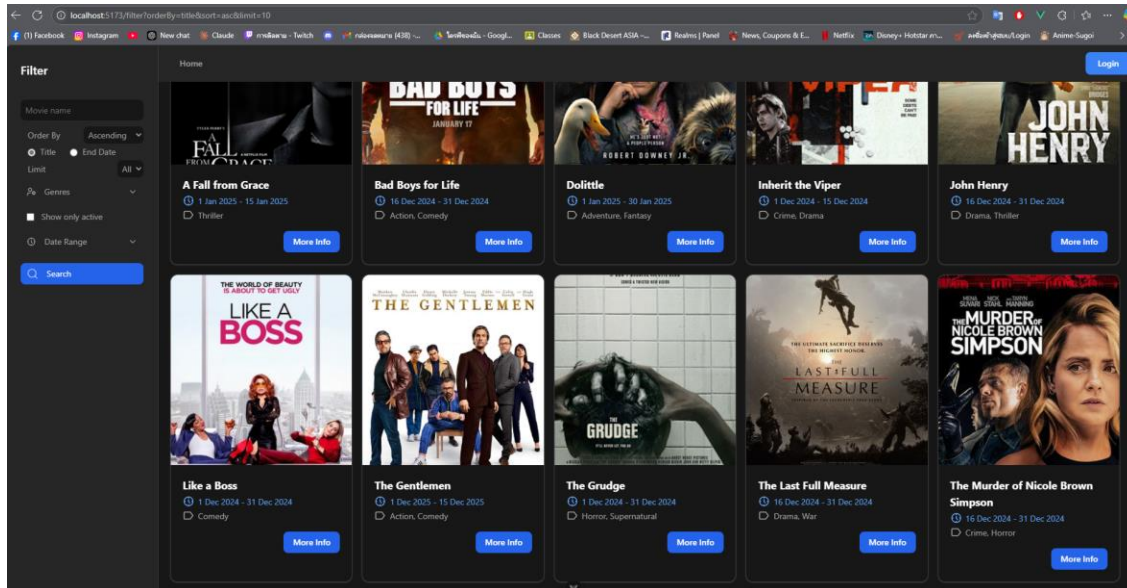
ผลลัพธ์



Code

```
28 async function getFilter() {
29   try {
30     moviesData.value = [];
31     isLoading.value = true;
32     let qry = null;
33     let movieRef = null;
34
35     movieRef = collection(db, "movies");
36
37     let filters = [];
38
39 >   if (route.query.genres) { ...
42   }
43
44 >   if (route.query.start && route.query.end) { ...
51   }
52
53   if (route.query.sort) {
54     if (route.query.sort === "asc") {
55       if (route.query.orderBy === "title") {
56         filters.push(orderBy("title", "asc"));
57       } else {
58         filters.push(orderBy("endDate", "asc"));
59       }
60     } else {
61       if (route.query.orderBy === "title") {
62         filters.push(orderBy("title", "desc"));
63       } else {
64         filters.push(orderBy("title", "desc"));
65       }
66     }
67   }
68
69 >   if (route.query.limit) { ...
73   }
74
75   qry = query(movieRef, ...filters);
76
77   if (qry) {
78     onSnapshot(qry, (querySnapshot) => {
79       moviesData.value = [];
80       querySnapshot.forEach((doc) => {
81         let data = doc.data();
82         data.id = doc.id;
83         if (route.query.movie) {
84           if (
85             data.title.toLowerCase().includes(route.query.movie.toLowerCase())
86           ) {
87             moviesData.value.push(data);
88           } else {
89             moviesData.value.push(data);
90           }
91         }
92       });
93       isLoading.value = false;
94     });
95   }
96 } catch (error) {
97   router.push({ name: "home" });
98 }
99 }
```

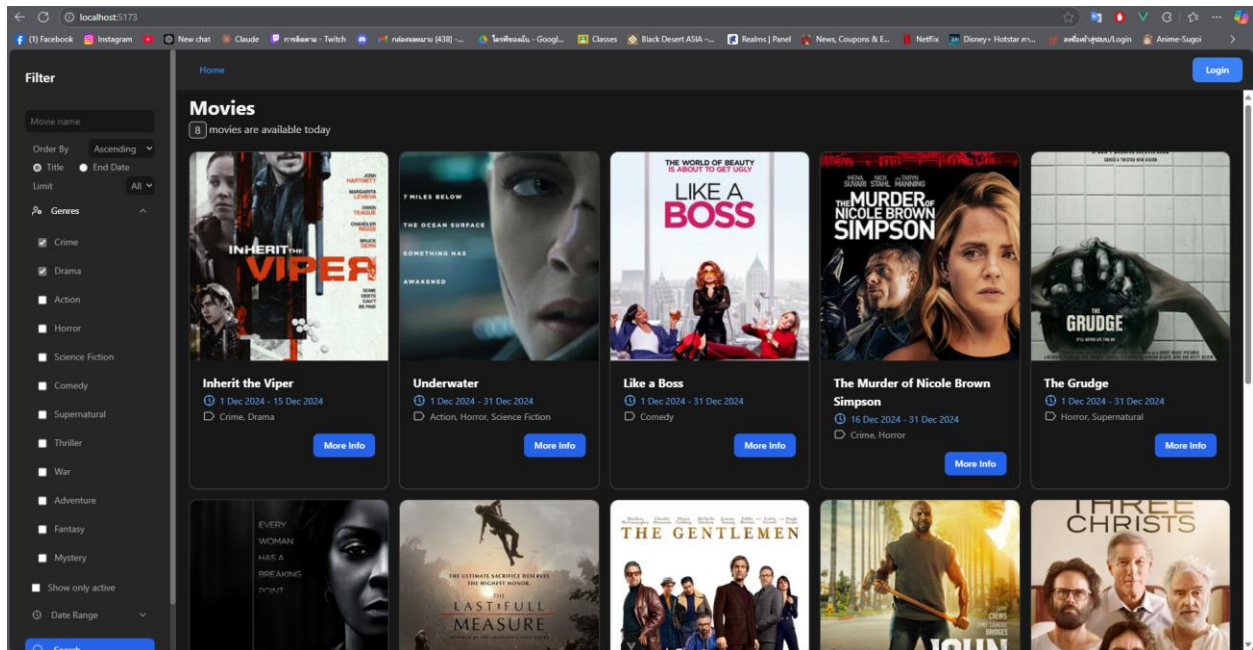
Filter ที่ 2 limit



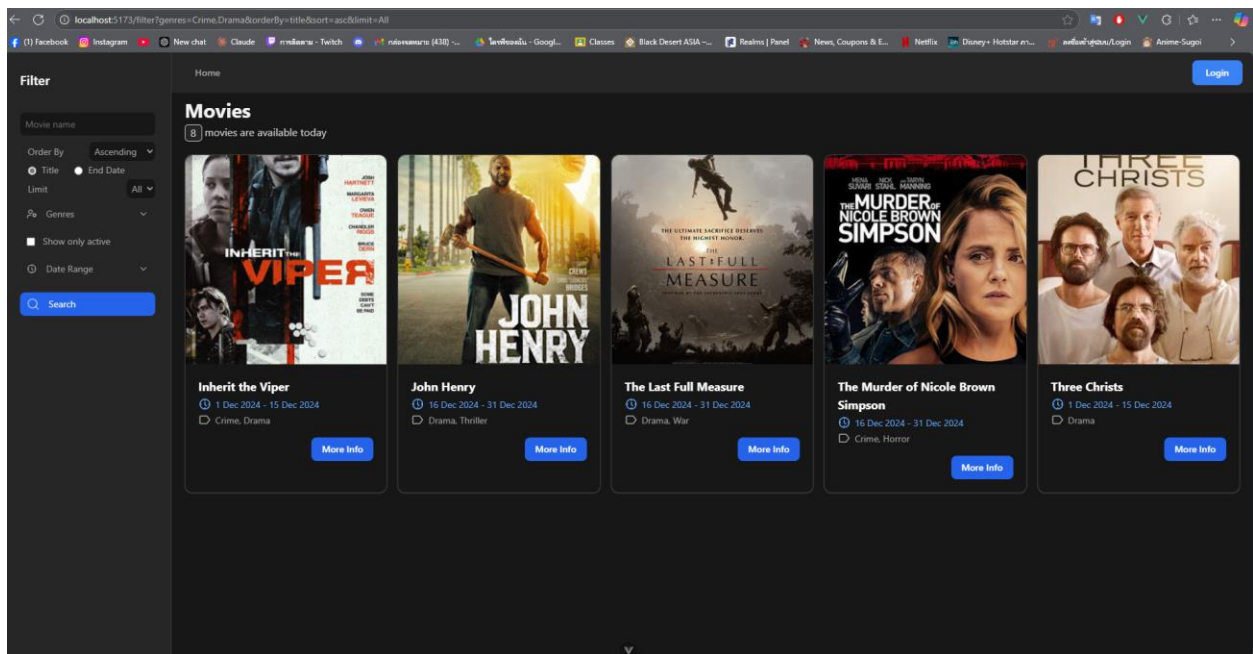
Code ถ้า parseInt ได้จะเป็น limit ตามตัวเลขนั้นๆ ไม่ได้ จะ All

```
28  async function getFilter() {
29    try {
30      moviesData.value = [];
31      isLoading.value = true;
32      let qry = null;
33      let movieRef = null;
34
35      movieRef = collection(db, "movies");
36
37      let filters = [];
38
69    if (route.query.limit) {
70      if (parseInt(route.query.limit)) {
71        filters.push(limit(parseInt(route.query.limit)));
72      }
73    }
74
75    qry = query(movieRef, ...filters);
76
77    if (qry) {
78      onSnapshot(qry, (querySnapshot) => {
79        moviesData.value = [];
80        querySnapshot.forEach((doc) => {
81          let data = doc.data();
82          data.id = doc.id;
83          if (route.query.movie) {
84            if (
85              data.title.toLowerCase().includes(route.query.movie.toLowerCase())
86            ) {
87              moviesData.value.push(data);
88            }
89          } else {
90            moviesData.value.push(data);
91          }
92        });
93        isLoading.value = false;
94      });
95    }
96  } catch (error) {
97    router.push({ name: "home" });
98  }
99 }
```


Filter ที่ 3 operators (array-contains-any)



ผลลัพธ์



Code

```
try {
  moviesData.value = [];
  isLoading.value = true;
  let qry = null;
  let movieRef = null;

  movieRef = collection(db, "movies");

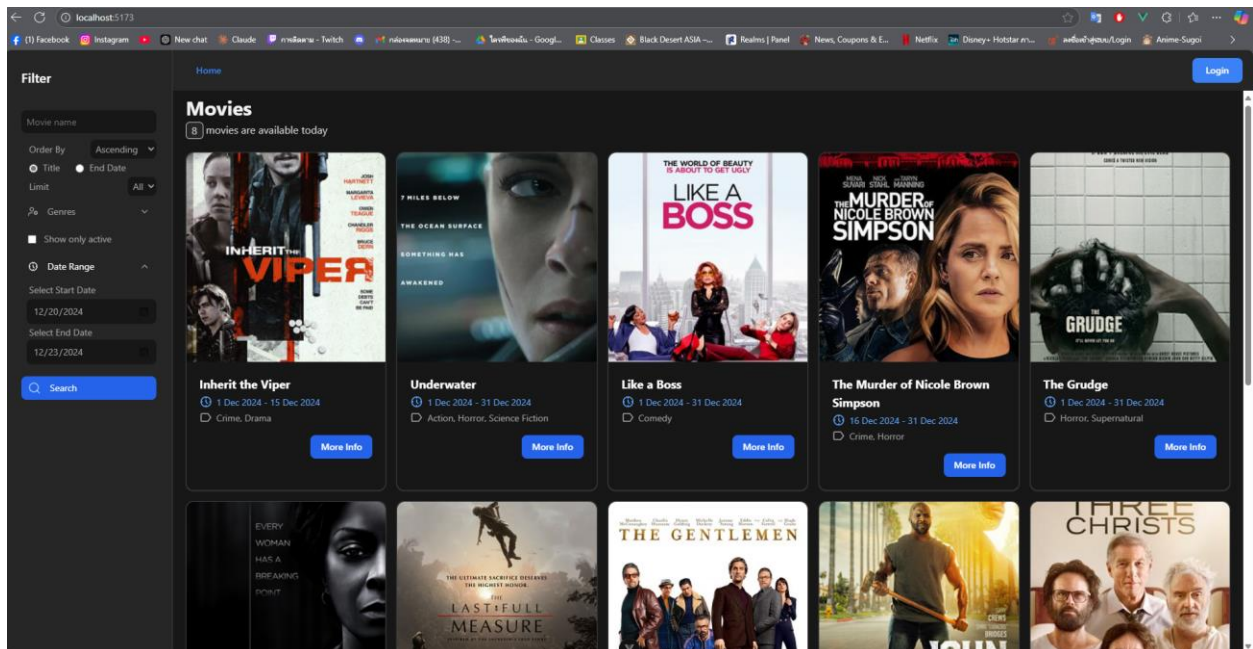
  let filters = [];

  if (route.query.genres) {
    const genres = route.query.genres.split(",");
    filters.push(where("genres", "array-contains-any", genres));
  }

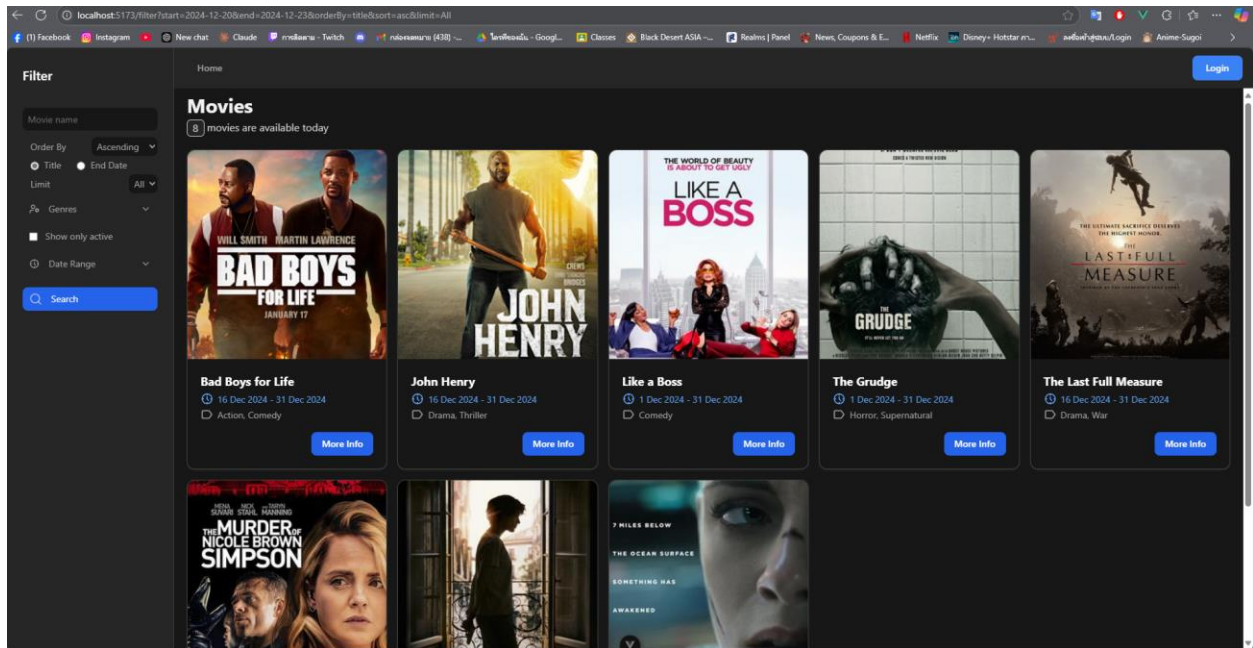
  qry = query(movieRef, ...filters);

  if (qry) {
    onSnapshot(qry, (querySnapshot) => {
      moviesData.value = [];
      querySnapshot.forEach((doc) => {
        let data = doc.data();
        data.id = doc.id;
        if (route.query.movie) {
          if (
            data.title.toLowerCase().includes(route.query.movie.toLowerCase())
          ) {
            moviesData.value.push(data);
          }
        } else {
          moviesData.value.push(data);
        }
      });
      isLoading.value = false;
    });
  }
} catch (error) {
  router.push({ name: "home" });
}
```

Filter ที่ 4 ranges เป็น list หนึ่งช่วงเวลาที่ต้องการชม



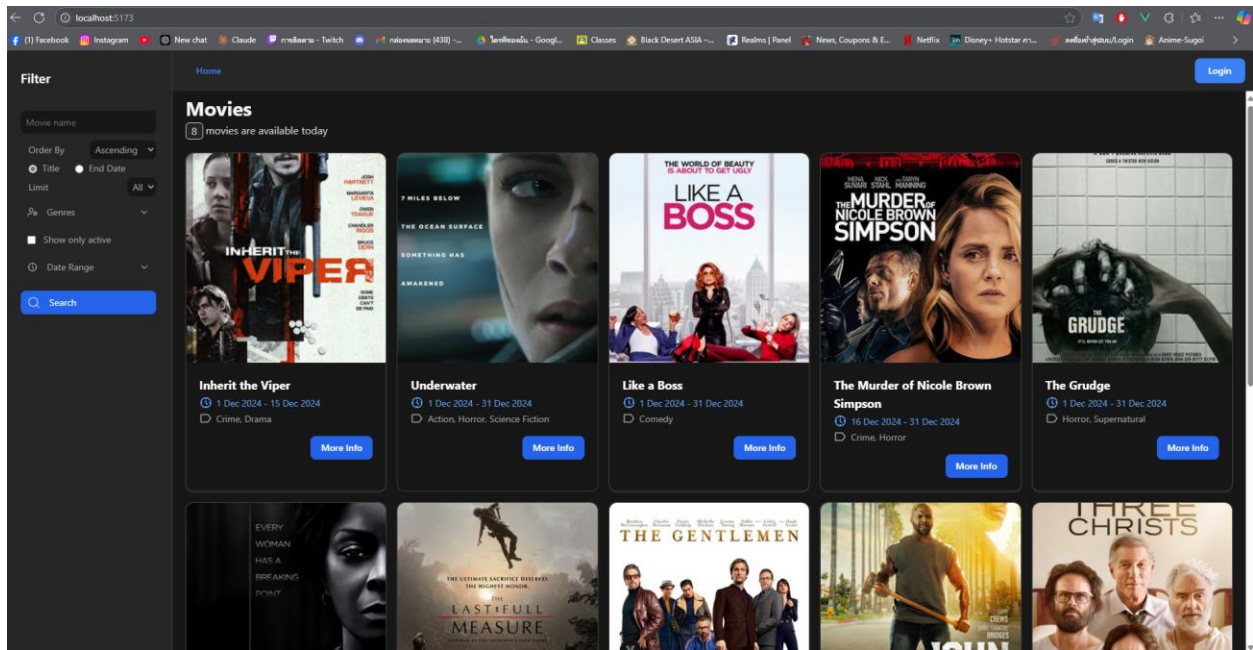
ผลลัพธ์



Code

```
28 async function getFilter() {
29   try {
30     moviesData.value = [];
31     isLoading.value = true;
32     let qry = null;
33     let movieRef = null;
34
35     movieRef = collection(db, "movies");
36
37     let filters = [];
38
39
40
41
42
43
44     if (route.query.start && route.query.end) {
45       const startDate = Timestamp.fromDate(new Date(route.query.start));
46       const endDate = Timestamp.fromDate(new Date(route.query.end));
47       filters.push(
48         where("startDate", "<=", endDate),
49         where("endDate", ">=", startDate)
50       );
51     }
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75   qry = query(movieRef, ...filters);
76
77   if (qry) {
78     onSnapshot(qry, (querySnapshot) => {
79       moviesData.value = [];
80       querySnapshot.forEach((doc) => {
81         let data = doc.data();
82         data.id = doc.id;
83         if (route.query.movie) {
84           if (
85             data.title.toLowerCase().includes(route.query.movie.toLowerCase())
86           ) {
87             moviesData.value.push(data);
88           }
89         } else {
90           moviesData.value.push(data);
91         }
92       });
93       isLoading.value = false;
94     });
95   }
96   catch (error) {
97     router.push({ name: "home" });
98   }
99 }
```

Query ที่ 5 เรื่อง aggregate เป็น จำนวน ภาพยนตร์มีให้ชมแล้ววันนี้



Code อยู่ใน Store

```
43   async getCountMovieCurrentDate() {
44     try {
45       const { startDate, endDate } = getToday();
46       const coll = collection(db, "movies");
47       const q = query(
48         coll,
49         where("startDate", "<=", endDate),
50         where("endDate", ">=", startDate)
51       );
52
53       const snapshot = await getAggregateFromServer(q, {
54         count: count(),
55       });
56       this.countMovieCurrentDate = snapshot.data().count;
57     } catch (error) {
58       console.error("Error getting movie count:", error);
59     }
60   },
61 },
62 });
```

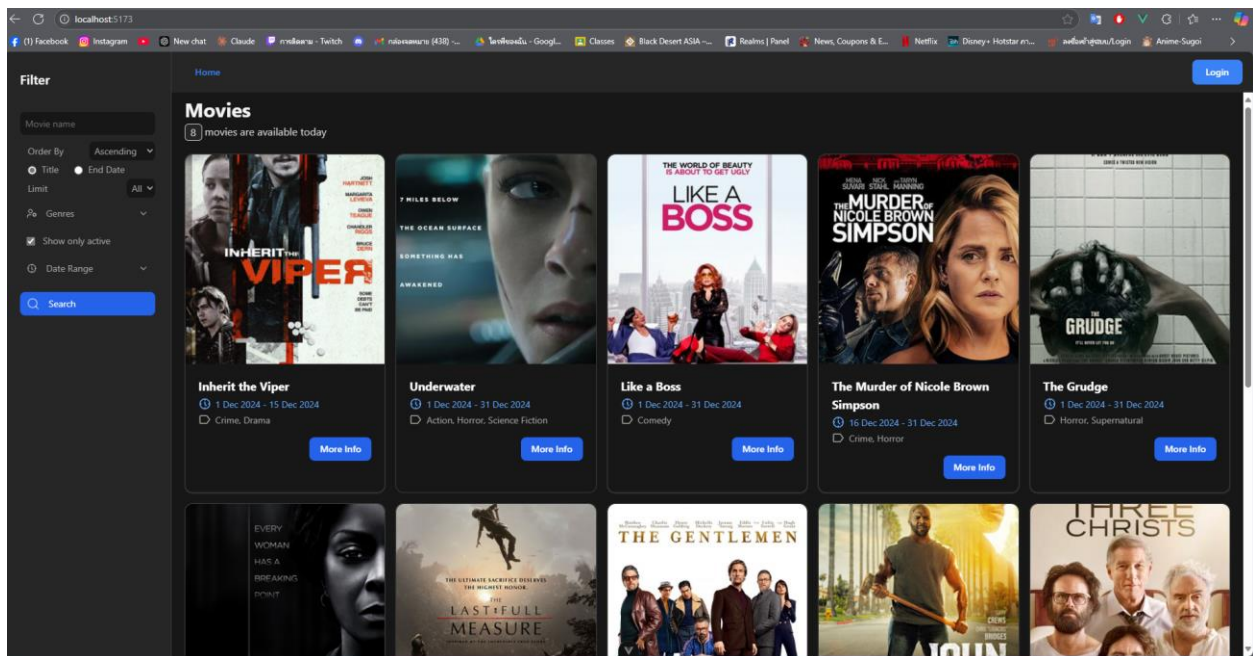
Utils getToday

```
function getToday() {
  const today = new Date();
  const startOfDay = new Date(today.setHours(0, 0, 0, 0)); // เวลาเริ่มต้นของวัน
  const endOfDay = new Date(today.setHours(23, 59, 59, 999)); // เวลาสิ้นสุดของวัน

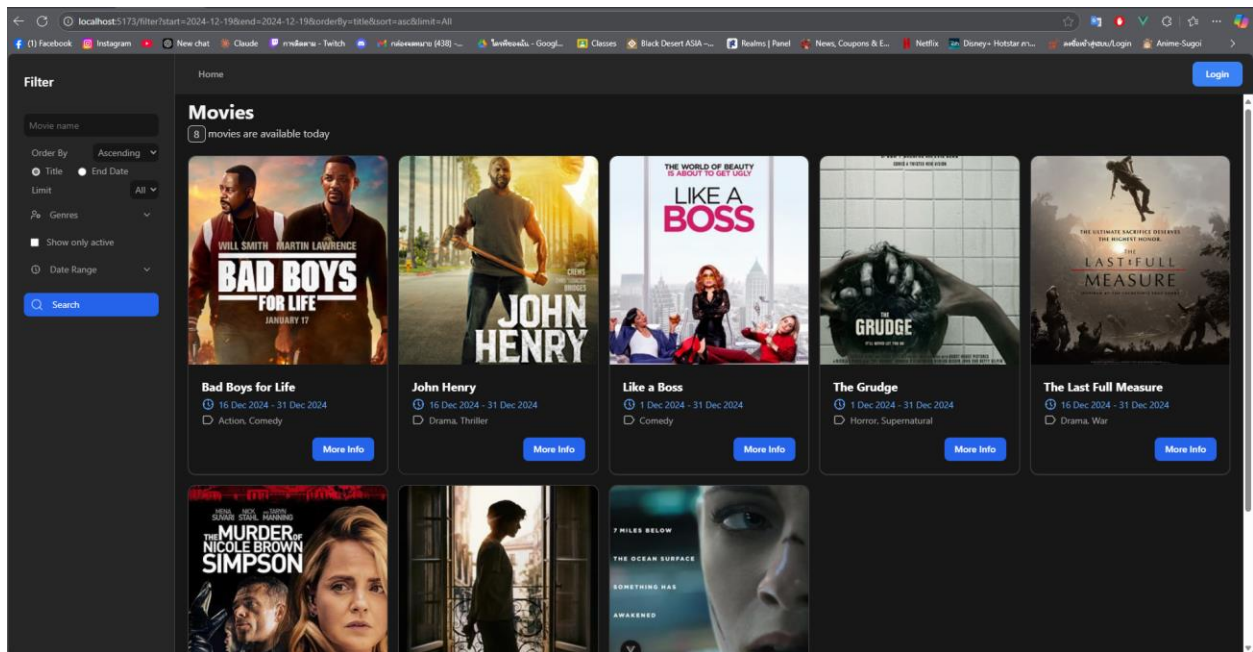
  const startDate = Timestamp.fromDate(startOfDay);
  const endDate = Timestamp.fromDate(endOfDay);

  return { startDate, endDate };
}
```

มีในส่วนของ Filter เรื่องนี้สำหรับดูหนังที่มีฉายวันนี้ กด Show only active และ Search




ผลลัพธ์



การจองตั๋วกด More Info และเลือกวันที่ต้องการจองและกดเวลา ของ Theater ที่ต้องการไปชม

Home

Login



Bad Boys for Life

Bad Boys for Life is a 2020 American buddy cop action comedy film directed by Adil & Bilal. It is the sequel to Bad Boys II (2003) and the third installment in the Bad Boys franchise. Will Smith, Martin Lawrence, Joe Pantoliano and Theresa Randle reprise their roles in the film and are joined by Paola Núñez, Vanessa Hudgens, Jacob Scipio, Alexander Ludwig, Charles Melton, Kate del Castillo and Nicky Jam. The film was produced by Smith, Jerry Bruckheimer, and Doug Belgrad, with a screenplay written by Chris Bremner, Peter Craig and Joe Carnahan. In Bad Boys for Life, Miami detectives Mike Lowrey and Marcus Burnett investigate a string...

16 Dec 2024 - 31 Dec 2024

Action, Comedy

Buy Tickets

Select Date : 20 Dec 2024

20 Dec 2024

21 Dec 2024

22 Dec 2024

23 Dec 2024

24 Dec 2024

25 Dec 2024

26 Dec 2024

27 Dec 2024

28 Dec 2024

29 Dec 2024

30 Dec 2024

31 Dec 2024

Select Theaters

Major Cineplex Central WestGate

Hall C

Thai

18:00:00

21:00:00

Major Cineplex Central World

Hall C

Thai

18:00:00

21:00:00

Major Cineplex Pinklao

Home

Login

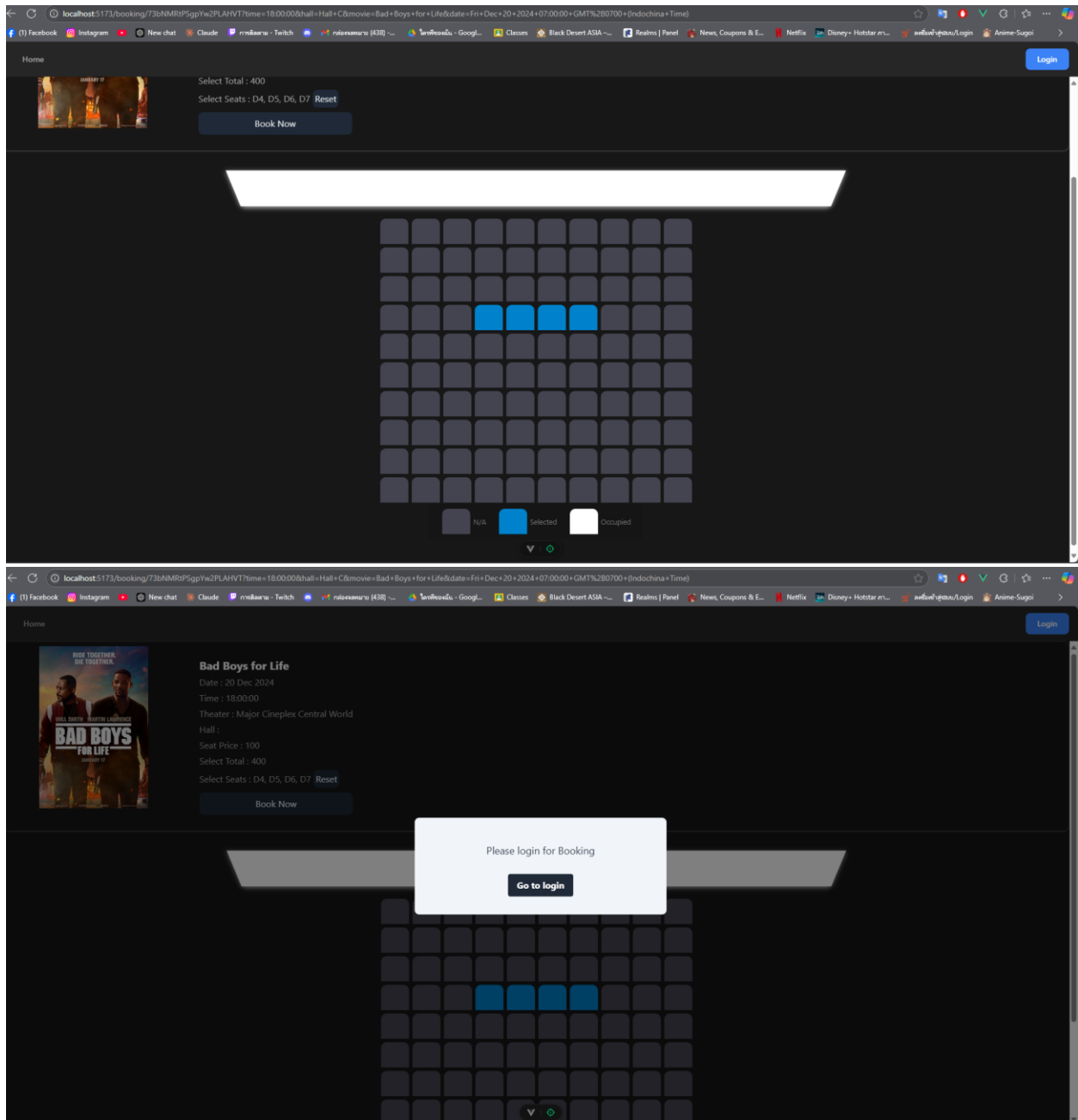


Bad Boys for Life

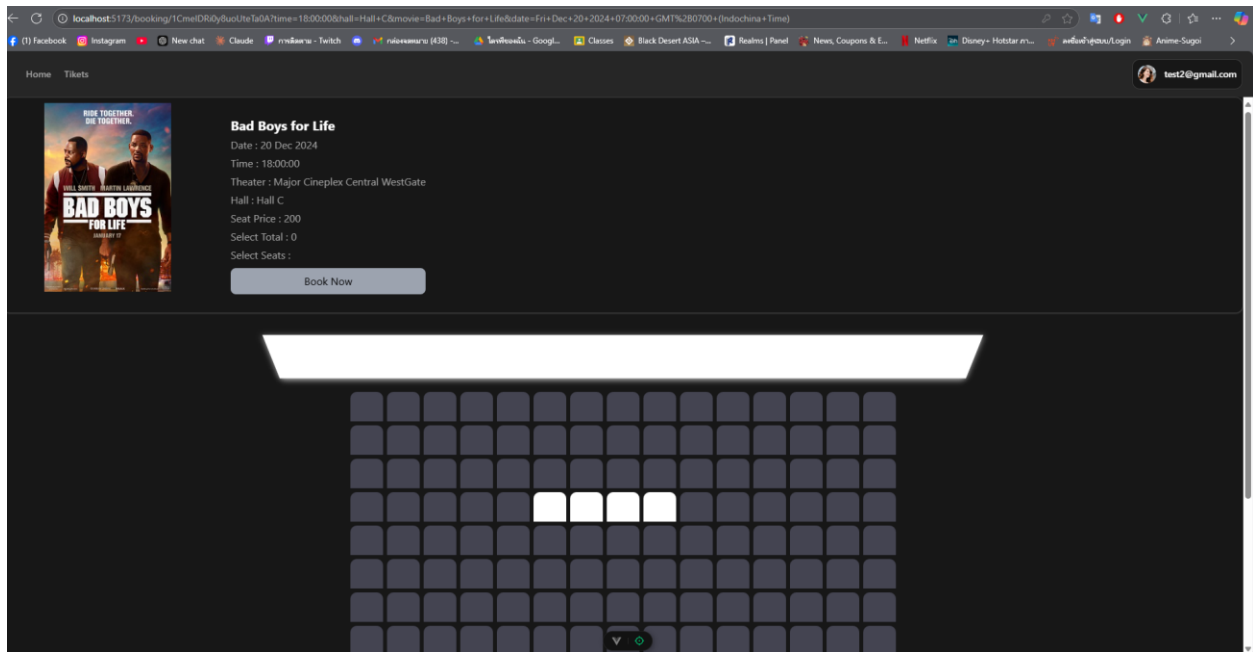
Date : 20 Dec 2024
Time : 18:00:00
Theater : Major Cineplex Central World
Hall :
Seat Price : 100
Select Total : 0
Select Seats :

Book Now

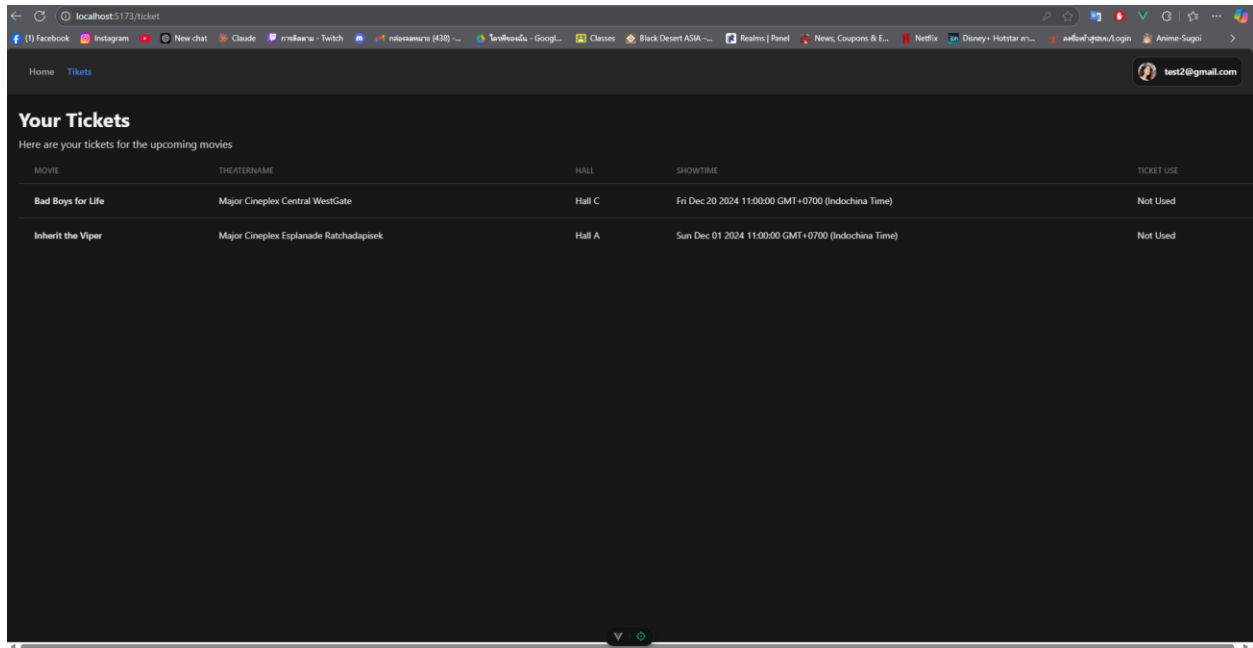
เลือกที่นั่ง กด Book Now ถ้าไม่ login จะมี Popup ให้ไป login ก่อน



ถ้าจองได้แล้วกลับมาหน้านี้ จะเห็นว่าจองได้แล้ว คนอื่นเข้ามาก็จะเห็นว่าจองแล้วจะเลือกไม่ได้



สามารถดู Tickets ได้



Login จะมีการไปหาข้อมูล ใน collection users ก่อน ถ้ามีจะเอาข้อมูล role และ email มาใช้

Code function login อยู่ใน Store ใช้ firebase/auth ในการ auth username password

```
35   async login(email, password) {
36     try {
37       const userCredential = await signInWithEmailAndPassword(
38         auth,
39         email,
40         password
41       );
42       const user = userCredential.user;
43       const userRef = doc(db, "users", user.uid);
44       const docSnap = await getDoc(userRef);
45       if (docSnap.exists()) {
46         this.user = user;
47         this.role = docSnap.data().role;
48         this.isAuthen = true;
49       } else {
50         throw new Error("User not found");
51       }
52     } catch (error) {
53       console.error("Login Error:", error.message);
54       throw error; // โยน Error กลับไปยังฟังก์ชันที่เรียกใช้งาน
55     }
56   },
57 }
```

Register จะใช้ firebase/auth ในการ Register username password และเอา user.uid มา set เป็น

Document ID และเก็บข้อมูลอื่นๆ

```
async register(email, password) {
  try {
    const userCredential = await createUserWithEmailAndPassword(
      auth,
      email,
      password
    );
    const user = userCredential.user;
    this.user = user;
    this.role = "user";
    await setDoc(doc(db, "users", user.uid), {
      email: user.email,
      role: this.role,
    });
  } catch (error) {
    console.error("Register Error:", error.message);
    throw error;
  }
},
```