

Τεχνικό Εγχειρίδιο για το Μοντέλο MVC και τη Χρήση της Βάσης Δεδομένων

Εισαγωγή

Το παρόν τεχνικό εγχειρίδιο αποτελεί έναν οδηγό για τη χρήση των αντικειμένων της Βάσης Δεδομένων μέσα στο πλαίσιο του μοντέλου MVC (Model-View-Controller). Η ανάλυση θα περιλαμβάνει παραδείγματα που εξηγούν τον τρόπο χρήσης και την αλληλεπίδραση με τα αντικείμενα της Βάσης Δεδομένων.

Περιεχόμενα

Τεχνικό Εγχειρίδιο για το Μοντέλο MVC και τη Χρήση της Βάσης Δεδομένων	1
Εισαγωγή.....	1
Κεφάλαιο 1: Επισκόπηση του Μοντέλου MVC	2
1.1 Μοντέλο (Model).....	2
1.2 Προβολή (View)	2
1.3 Ελεγκτής (Controller)	4
1.4 Συνεργασία Μοντέλου, Προβολής και Ελεγκτή	4
Κεφάλαιο 2: Αντικείμενα της Βάσης Δεδομένων	4
2.1 Users.....	5
2.2 Αντικείμενα σχετικά με τα σινεμά.....	6
Κεφάλαιο 3: Παραδείγματα Εφαρμογής	6
3.1 Φόρμα Σύνδεσης (Login)	6
3.2 Φόρμα Κράτησης Εισιτηρίων.....	7
3.3 Λίστα Κρατήσεων	8
3.4 Προβολή Ταινιών.....	8

Κεφάλαιο 1: Επισκόπηση του Μοντέλου MVC

Στο πλαίσιο αυτού του εγγράφου, θα εξετάσουμε το μοντέλο MVC σε σχέση με τα παρακάτω μοντέλα της Βάσης Δεδομένων, που υλοποιούνται στην εφαρμογή:

1.1 Μοντέλο (Model)

Η εφαρμογή αποτελείται από διάφορες κλάσεις που αντιπροσωπεύουν τις κύριες οντότητες του συστήματος. Κάθε κλάση παρουσιάζει μια συγκεκριμένη οντότητα με τα δικά της χαρακτηριστικά. Στην παρούσα υλοποίηση, οι κύριες κλάσεις περιλαμβάνουν τις εξής:

- ❖ **User**: Αντιπροσωπεύει έναν χρήστη του συστήματος.
- ❖ **Admin**: Αναπαριστά ένα διαχειριστή.
- ❖ **ContentAdmin**: Αναπαριστά έναν διαχειριστή περιεχομένου.
- ❖ **Customer**: Αντιπροσωπεύει έναν πελάτη.
- ❖ **Movie**: Παριστάνει μια ταινία στο σύστημα.
- ❖ **Cinema**: Αντιπροσωπεύει μια αίθουσα προβολής στον κινηματογράφο.
- ❖ **Provoli**: Παριστάνει μια προβολή ταινίας σε συγκεκριμένη αίθουσα.
- ❖ **Reservation**: Καταγράφει μια κράτηση από έναν πελάτη για συγκεκριμένη προβολή.

1.2 Προβολή (View)

Στο πλαίσιο της αρχιτεκτονικής MVC, οι προβολές είναι υπεύθυνες για την εμφάνιση των δεδομένων στους χρήστες. Οι προβολές συνδέονται με το μοντέλο και εμφανίζουν τα δεδομένα με τρόπο που καθορίζεται από τον ελεγκτή.

- ❖ **Account**: όλα τα views σχετίζονται με τους χρήστες και τους λογαριασμούς αυτών:
 - **Create**: έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί δημιουργήσει έναν καινούργιο χρήστη της εφαρμογής.
 - **Delete**: έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί να διαγράψει έναν χρήστη.
 - **Details**: έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορούν να προβληθούν τα στοιχεία ενός χρήστη.
 - **Edit**: έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί να επεξεργαστεί τα στοιχεία ενός χρήστη.
 - **Index**: έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορούν να θεαθούν όλοι οι χρήστες της εφαρμογής.
 - **Login**: οι εγγεγραμμένοι χρήστες αποκτούν πρόσβαση στην εφαρμογή μέσω σύνδεσης.
 - **Register**: εκπονεί χρέη πύλης καθώς αποτελεί την πλατφόρμα εγγραφής.
- ❖ **Cinema**: τα views πραγματοποιούν λειτουργίες που σχετίζονται με τα διαθέσιμα σινεμά.

- **Details:** προβολή πληροφοριών ενός σινεμά καθώς και ποιες ταινίες παίζονται σε αυτό.
- **Index:** λίστα με όλα τα σινεμά.
- ❖ **ContentAdmins:** τα view εκπονούν λειτουργίες είτε για έναν διαχειριστή περιεχομένου (content admin) είτε που της πραγματοποιεί ένας διαχειριστής περιεχομένου (content admin).
 - **Create:** έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί δημιουργήσει έναν καινούργιο διαχειριστή περιεχομένου (content admin).
 - **CreateMovie:** έχει πρόσβαση μόνο ένας διαχειριστή περιεχομένου (content admin), και μπορεί να προσθέσει μία νέα ταινία.
 - **CreateProvoli:** έχει πρόσβαση μόνο ένας διαχειριστή περιεχομένου (content admin), και μπορεί να προσθέσει μία νέα προβολή ταινίας.
 - **Delete:** έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί διαγράψει έναν διαχειριστή περιεχομένου (content admin).
 - **Details:** έχει πρόσβαση μόνο ένας διαχειριστής(admin), εμφανίζονται όλες οι πληροφορίες σχετικά με κάποιο διαχειριστή περιεχομένου (content admin).
 - **Edit:** έχει πρόσβαση μόνο ένας διαχειριστής(admin), και μπορεί επεξεργαστεί τα στοιχεία ενός διαχειριστή περιεχομένου (content admin).
 - **EditProvoli:** έχει πρόσβαση μόνο ένας διαχειριστή περιεχομένου (content admin), και μπορεί να επεξεργαστεί μία προβολή ταινίας.
 - **Index:** έχει πρόσβαση μόνο ένας διαχειριστής(admin), λίστα όλων των διαχειριστή περιεχομένου (content admin).
- ❖ **Home:** αποτελεί τις κεντρικές σελίδες της εφαρμογής
 - **Index:** αρχική σελίδα
- ❖ **Movie:** τα views πραγματοποιούν λειτουργίες που σχετίζονται με τις ταινίες.
 - **Details:** προβολή πληροφοριών μιας ταινίας καθώς και λίστα με τα σινεμά που μπορεί να παίζεται η ταινία.
 - **Index:** λίστα με όλα τις ταινίες.
- ❖ **Provoli:** τα views πραγματοποιούν λειτουργίες που σχετίζονται με τις διαθέσιμες προβολές.
 - **Index:** παρουσιάζονται οι πληροφορίες για τις προβολές για την επιλεγμένη ταινία και το επιλεγμένο σινεμά.
 - **Notfound:** ενημέρωση ότι δεν βρέθηκε καμία προβολή για την επιλεγμένη ταινία και το επιλεγμένο σινεμά.
- ❖ **Reservation:** τα views πραγματοποιούν λειτουργίες που σχετίζονται με τις κρατήσεις εισιτηρίων.
 - **Create:** κράτηση εισιτηρίων, έχει πρόσβαση μόνο ένας πελάτης (customer).
 - **History:** προβολή ιστορικού κρατήσεων εισιτηρίων, έχει πρόσβαση μόνο ένας πελάτης (customer).
- ❖ **Shared:** αποτελείτε κυρίως από partial views που χρησιμοποιούνται κατά μήκος όλης της εφαρμογής.

- **_Layout**: αποτελεί συνδέσεις με css και javascript, καθώς περιέχει την γραμμή περιήγησης (navbar) της εφαρμογής.
- **_ValidationScriptsPartial**: περιέχει συνδέσεις με javascript οι οποίες ελέγχουν την ταυτοποίηση και την ορθότητα των στοιχείων που εισέρχονται στην εφαρμογή.

1.3 Ελεγκτής (Controller)

Ο ελεγκτής λειτουργεί ως διαμεσολαβητής μεταξύ του μοντέλου και της προβολής. Αναλαμβάνει τη διαχείριση των ενεργειών του χρήστη, αλληλεπιδρά με το μοντέλο για την ανάκτηση και ενημέρωση δεδομένων, και στη συνέχεια καθοδηγεί την προβολή για την εμφάνιση των αποτελεσμάτων.

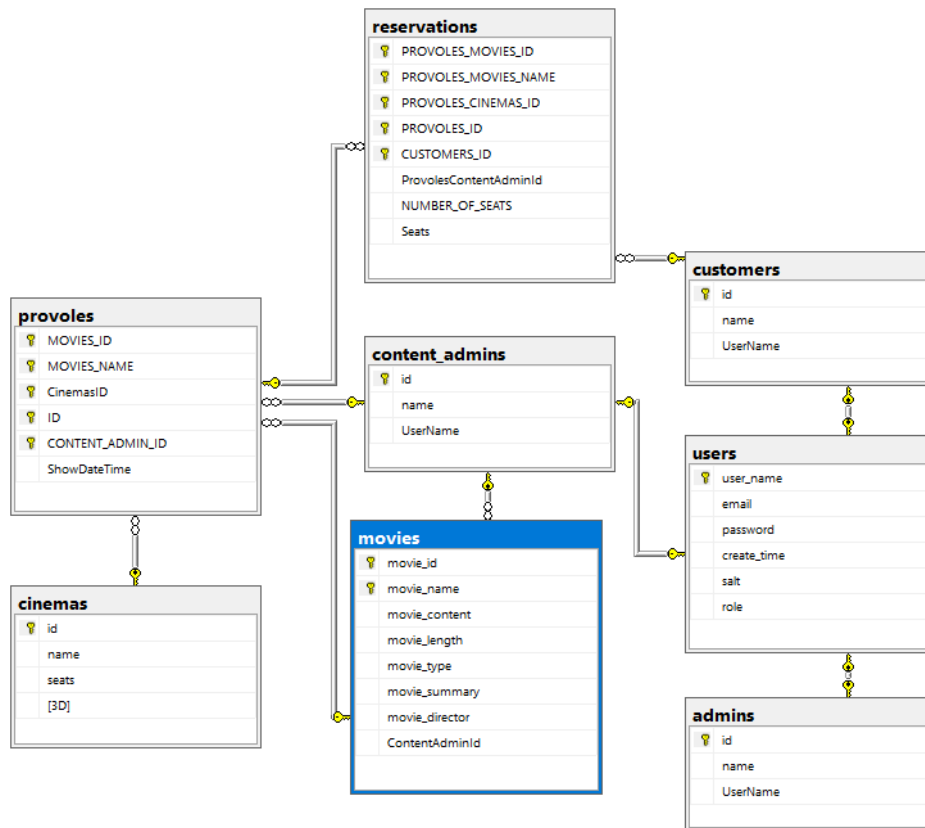
- ❖ **AccountController**: περιέχει λειτουργίες που σχετίζονται με τους χρήστες και τους λογαριασμούς αυτών.
- ❖ **CinemaController**: περιέχει λειτουργίες που σχετίζονται με τα cinema.
- ❖ **ContentAdminsController**: περιέχει λειτουργίες είτε για έναν διαχειριστή περιεχομένου (content admin) είτε που της πραγματοποιεί ένας διαχειριστής περιεχομένου (content admin).
- ❖ **HomeController**: βοηθάει στην εμφάνιση των κεντρικών σελίδων της εφαρμογής.
- ❖ **MovieController**: περιέχει λειτουργίες που σχετίζονται με τις ταινίες.
- ❖ **ProvoliController**: περιέχει λειτουργίες που σχετίζονται με τις προβολές.
- ❖ **ReservationController**: περιέχει λειτουργίες που σχετίζονται με την προβολή αλλά και την κράτηση εισιτηρίων.

1.4 Συνεργασία Μοντέλου, Προβολής και Ελεγκτή

Η συνεργασία μεταξύ των τριών συνιστωσών είναι κρίσιμη για την αποτελεσματική λειτουργία του συστήματος. Ο χρήστης αλληλεπιδρά με το σύστημα μέσω του ελεγκτή, ο οποίος διαμορφώνει τα αιτήματα προς το μοντέλο. Το μοντέλο αποκτά και ενημερώνει τα δεδομένα, και ο ελεγκτής καθοδηγεί την προβολή για την εμφάνιση των αποτελεσμάτων.

Κεφάλαιο 2: Αντικείμενα της Βάσης Δεδομένων

Σε αυτό το κεφάλαιο, θα αναλυθούν τα αντικείμενα της Βάσης Δεδομένων που χρησιμοποιούνται. Θα παρουσιαστούν παραδείγματα που εξηγούν τον τρόπο χρήσης τους και τις βασικές λειτουργίες που προσφέρουν.



Εικόνα 1: διάγραμμα αντικειμένων βάσης δεδομένων.

2.1 Users

Εδώ θα αναλυθούν οι χρήστες της εφαρμογής, καθώς και οι τύποι τους, παρέχοντας παραδείγματα και περιγραφές χρήσης.

- ❖ **Users:** περιλαμβάνονται όλοι οι χρήστες της εφαρμογής και αποθηκεύονται στοιχεία τους όπως: username, email, password (κωδικοί), create-time (ημερομηνία δημιουργίας χρήστη), salt (χρησιμοποιείτε για μεγαλύτερη ασφάλεια των κωδικών), role (ο ρόλος κάθε χρήστη)

Τύποι χρηστών, και οι τρεις τύποι χρηστών αποθηκεύουν στην βάση id, name (όνομα) και ως foreign key το username του σχετιζόμενου χρήστη.

- ❖ **Admins:** διαχειριστές, μπορούν να δημιουργού και να επεξεργάζονται χρήστες και τους τύπους των χρηστών.
- ❖ **Content Admins:** διαχειριστές περιεχομένου, προσθέτουν και επεξεργάζονται περιεχόμενο όπως ταινίες, προβολές και σινεμά.
- ❖ **Customers:** απλοί χρήστες της εφαρμογής βλέπουν τις διαθέσιμες ταινίες και σινεμά και μπορούν να μεταβούν σε κράτηση εισιτηρίων για συγκεκριμένες προβολές.

2.2 Αντικείμενα σχετικά με τα σινεμά

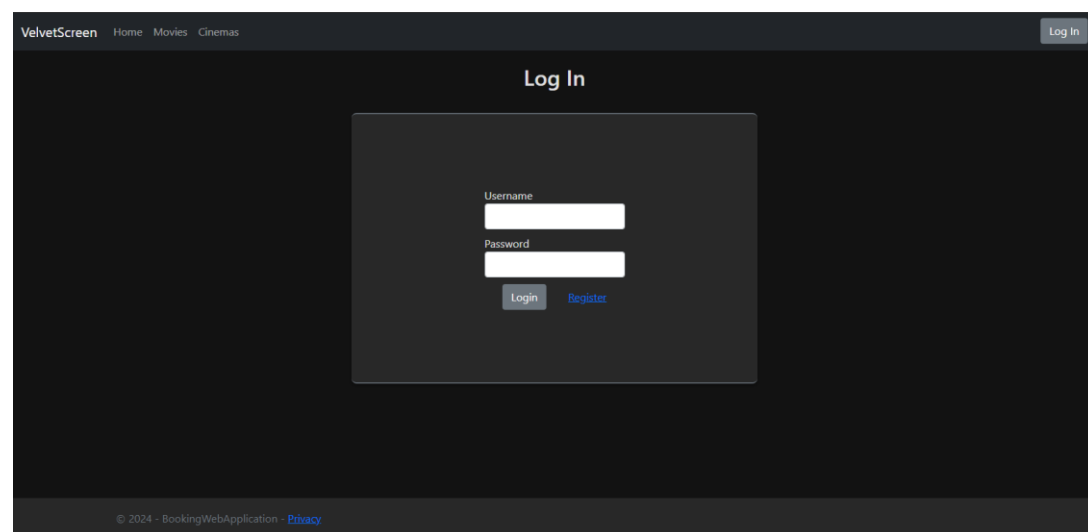
Σε αυτή την ενότητα, θα εξεταστούν τα αντικείμενα που σχετίζονται με τα σινεμά, παρέχοντας περαιτέρω λεπτομέρειες και παραδείγματα χρήσης. Αυτά αποτελούν τα ίδια τα σινεμά, τις ταινίες, τις προβολές αλλά και τις κρατήσεις εισιτηρίων.

- ❖ **Cinemas**: τα σινεμά αναπαριστούν τους πραγματικούς χώρους των σινεμά, αποθηκεύοντας πληροφορίες όπως το id, το όνομα του σινεμά, των αριθμό των θέσεων των αιθουσών και αν περιλαμβάνουν 3D (τρισεδιάστατες) δυνατότητες.
- ❖ **Movies**: οι ταινίες περιέχουν όλες τις πληροφορίες που μπορεί να είναι χρήσιμες για να αποφασίσει ένας χρήστης ποια θα επιλέξει. Περιλαμβάνουν: το id, το όνομα της ταινίας, το content αυτής, το μήκος της, τον τύπο της (genre), μία περίληψη της ταινίας, το όνομα του παραγωγού καθώς και το ποιος content admin την πρόσθεσε.
- ❖ **Provoles**: οι προβολές ενημερώνουν το χρήστη για το πότε και σε ποιο σινεμά παίζεται μία ταινία. Περιλαμβάνουν: το id της ταινία, το όνομα της, το id του σινεμά, ένα γενικό id, το id του content admin που την πρόσθεσε και την ημερομηνία και ώρα προβολής της ταινίας.
- ❖ **Reservations**: οι κρατήσεις αποτελούν τις πληροφορίες για την κράτηση εισιτηρίων που έκανε κάποιος πελάτης (customer). Περιλαμβάνουν: το id της ταινίας, το όνομα αυτής, το id του σινεμά, το id της προβολής, το id του πελάτη, τον αριθμό των θέσεων που έκλεισε και ποιες είναι αυτές οι θέσεις.

Κεφάλαιο 3: Παραδείγματα Εφαρμογής

Σε αυτό το κεφάλαιο, θα παρουσιαστούν πρακτικά παραδείγματα που δείχνουν τη συνεργασία των αντικειμένων της Βάσης Δεδομένων με άλλα μέρη του μοντέλου MVC.

3.1 Φόρμα Σύνδεσης (Login)



Εικόνα 2: φόρμα σύνδεσης.

```

<div class="login-box">

    <form asp-action="Login" method="post">

        <label for="Username">Username</label>

        <input class="login-control form-control" type="text" name="Username" required />

        <label for="Password">Password</label>

        <input class="login-control form-control" type="password" name="Password" required

    />

    <div class="login-buttons-links">

        <button class="login-button btn btn-secondary" type="submit">Login</button>

        <a class="register-link" asp-action="Register">Register</a>

    </div>

</form>

</div>

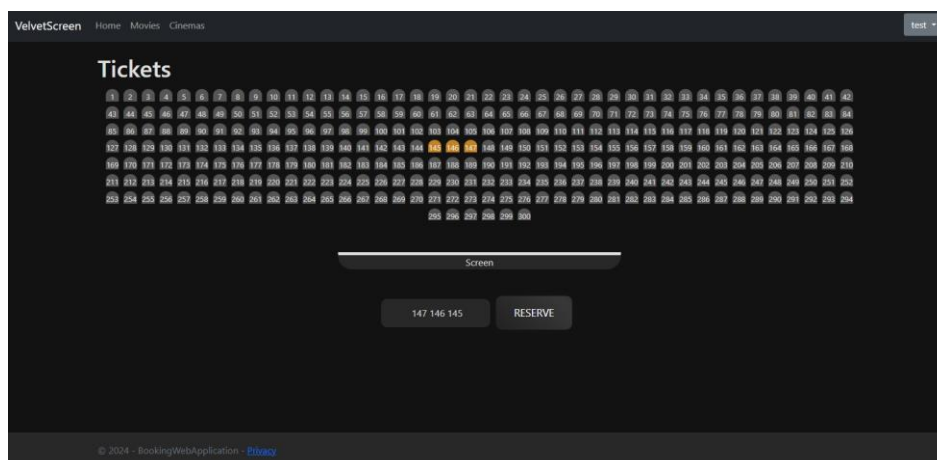
```

Εικόνα 3: html φόρμα σύνδεσης.

Σε αυτό το παράδειγμα, ο χρήστης καλείται να εισάγει το όνομα χρήστη του στο πεδίο "Username" και τον κωδικό πρόσβασής του στο πεδίο "Password". Το "asp-action" και "method" της φόρμας πρέπει να αντιστοιχούν στις διαδρομές και τις μεθόδους που χρησιμοποιούνται στον AccountController για την επεξεργασία των συνδέσεων.

Όταν ο χρήστης πατήσει το κουμπί "Login," η φόρμα θα στείλει τα δεδομένα στον AccountController, όπου μπορεί να γίνει ο έλεγχος της ταυτότητας και η επικύρωση του χρήστη.

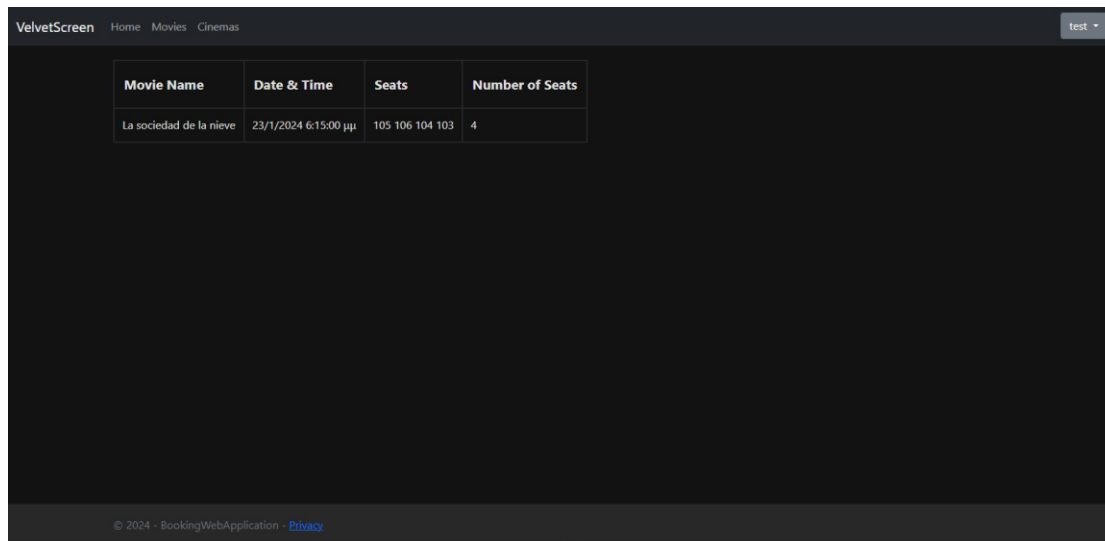
3.2 Φόρμα Κράτησης Εισιτηρίων



Εικόνα 4: φόρμα κράτησης εισιτηρίων.

Ένας χρήστης μπορεί να χρησιμοποιήσει μια φόρμα για να επιλέξει την ταινία, την ημερομηνία, την ώρα και τις θέσεις που θέλει να κρατήσει.

3.3 Λίστα Κρατήσεων



The screenshot shows a web application interface with a dark theme. At the top, there is a navigation bar with the text 'VelvetScreen' and links for 'Home', 'Movies', and 'Cinemas'. A 'test' button is visible in the top right corner. Below the navigation bar, there is a table with the following data:

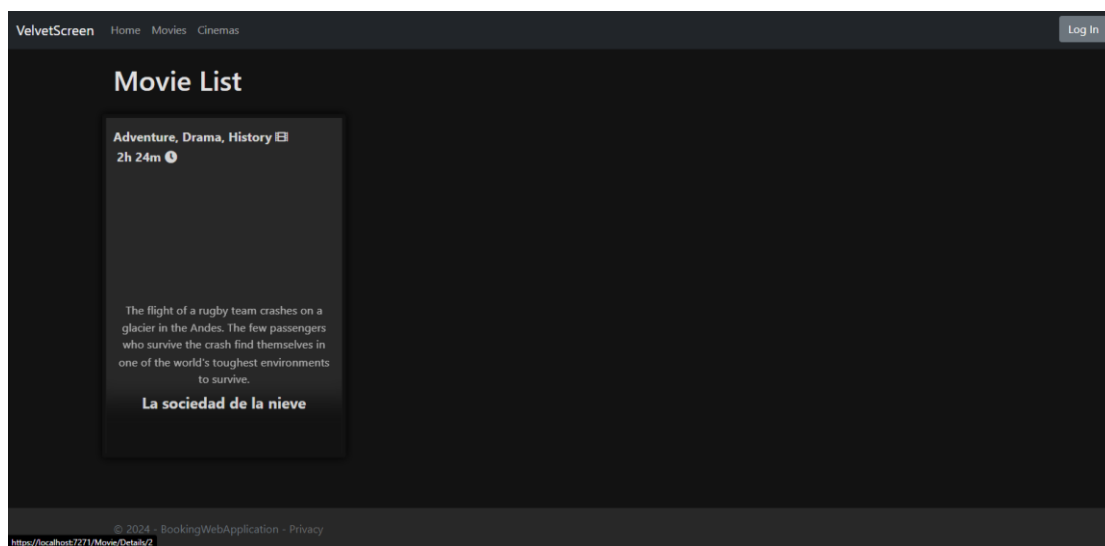
Movie Name	Date & Time	Seats	Number of Seats
La sociedad de la nieve	23/1/2024 6:15:00 μμ	105 106 104 103	4

At the bottom of the page, there is a footer with the text '© 2024 - BookingWebApplication - Privacy'.

Εικόνα 5: λίστα ιστορικού κρατήσεων εισιτηρίων.

Ο χρήστης μπορεί να δει μια λίστα με τις προηγούμενες κρατήσεις του.

3.4 Προβολή Ταινιών



Εικόνα 6: προβολή λίστας ταινιών

```
public async Task<IActionResult> Index()
{
    return View(await _dbContext.Movies.ToListAsync());
}
```

Εικόνα 7: Movie Controller get every movie and send it to the view

Ο Movie Controller παίρνει όλες τις ταινίες από την βάση και τις στέλνει στο Movie < Index view όπου και παρουσιάζονται με μορφή κάρτας

```
<div class="centered">

  <section class="cards">

    @foreach (var item in Model.OrderBy(e => e.MovieName))
    {

      <a asp-action="Details" asp-route-id="@item.MovieId">

        <div class="card">

          <div class="title">@item.MovieName</div>

          <div class="text">@item.MovieSummary</div>

          <div class="type">@item.MovieType <i class="fas fa-film"></i></div>

          <div class="length">@string.Format("{0}h {1}m", item.MovieLength / 60,
item.MovieLength % 60) <i class="fas fa-clock"></i></div>

        </div>

      </a>

    }

  </section>

</div>
```

Εικόνα 8: παρουσίαση ταινίες σε μορφή κάρτας