**BUILDING A BOOKING EVENT WEBSITTE**

**APPLYING TO REACTJS AND SPRING BOOT**

**INTRODUCTION**

## **1. The urgency of project**

With the explosion of technology, people's needs and consumer behavior change. Digital transformation is essential for most businesses. Therefore, in order to catch up with customer trends, businesses all require business models on mobile applications or websites… Specifically in this topic is booking event tickets based on a web platform.

Prior to the development of technology, Most of the events that customers want to attend must go to the venue or authorized by the organizer to buy tickets directly. Understanding the difficulties of both parties, A website platform to mediate for both customers and organizers to meet the requirement of both parties is essential. And that is also the reason why we developed the topic "Building an event ticketing system".

With the features of this website, customers will have the opportunity to buy tickets online instead of directly as before, along with transparent and clear information publicly available on the platform, where there is censorship of the team prior to officially allowing the organizer to operate with the aim of creating credibility and reliability when buying tickets on the online platform. In addition, the Organizer can easily manage their events and be flexible in the process of implementing the event, or when notifying customers of an emergency situation.

## **2. Research objects**

For this topic, the research object is based on user needs, referring to related websites and models to create an intermediary transaction place between ticket sellers and buyers. In addition, the technologies applied to realize specific products are as follows:

- Restful API application to build APIs to support accessing and manipulating the system between client and server effectively.

- Using NoSQL database is MongoDB to store data in the system.

- For UI rendering, the team uses the ReactJS framework, AntDesign, Mui, Bootstrap and other related support libraries to build the interface on the website.

- Regarding system security, conducting research on Json Web Token technology

make the system secure and all requests are controlled via Token.

**3. Research scope**

The website mainly focuses on handling the basic operations of a website as an intermediary platform in buying and selling event tickets with key functions such as: account management, list management, etc. favorite events, view event details, search for events by various methods

## **4. Project objectives**

Topic “BUILDING A BOOKING EVENT WEBSITE APPLYING TO REACTJS AND SPRING BOOT” is including purposes as follows:

- Building a website that allows users to search, view information and order event tickets.

- Building a management system and statistics on sales of orders and events for businesses.

- The website has a user-friendly interface and is secure for users' personal information.

**CHAPTER 2: PRESENT SURVEY AND USER REQUIREMENT**

* 1. **Present survey**

<https://ticketbox.vn/>

Graphical user interface, website

Description automatically generated

*Figure 2‑1. Ticketbox*

-Main function:

-Viewing prominent event lists and a variety of events

-Allowing user to purchase tickets or organize events

-Finding appropriate events based on address, date, ticket costs.

-Viewing information of purchased tickets or administering events (if any)

-Advantages:

-Having diverse events

-Transparent information after buying ticket(s)

-Having a reliable and authentic website

-Paying order with different methods

-Send email when buying ticket successfully

-Disadvantages:

-User interface is relatively complex for user

-Depending on support team to verify a event

-Can not see exactly quantity of Ticket when placing order

<https://ticketgo.vn/>

Graphical user interface, website

Description automatically generated

*Figure 2‑2. TicketGo*

-Main function:

-Viewing from prominent events to normal ones from top to bottom

-Viewing weekly organized events

-Booking events through hotline number affiliated with organization

-Advantages:

-Transparent event information

-Having diverse events

- Paying order with different methods

-Disadvantages:

-Viewing blurry word font

-User interface is not optimal about spacing screening.

-Having leftover server errors

<https://introart.com.vn/>

Graphical user interface

Description automatically generated

*Figure 2‑3. IntroArt*

-Main function:

-Viewing events based on week/month

-Viewing specific event(s)

-Booking ticket(s) through hotline number affiliated with organization

-Advantages:

-Having good of user interface

-Having appropriate layout

-Disadvantages:

-Rendering low-performance user interface

* 1. **Determine requirements**

**2.2.1 Functional requirements**

-Store data:

o Save avatar image of user

o Save List of events is holding/completed

o Save personal information for the purpose of contact

- Management:

o Update Profile

o Add/Delete/Update Event

o Add/Delete/Update favorite event

o Approve/Refuse pending Organizer Account

- Search:

o Search by name

o Search by filters

- Statistic:

o Statistics group by Order

o Statistics group by Ticket

o Statistics group by Event

-Authorization:

o Provide user with equivalent authorization to access to url accordingly

-Authentication

o Allow user to log in with equivalent role

**2.2.2 Non functional requirements**

-Searching information faster and more effective according to user requirement

-Provide user with equivalent authorization

-Ensure that securing personal information is private

-Minimizing the possibility of leaking private data.

-Processing and responding data between client side and back-end side is on time and quickly

-Enhancing user experience

-Having possibility of expanding functions depending on user requirement

**2.3.3 User requirement**

**2.3.3.1. Actor of system**

-Admin: : Use the system with the highest rank. This individuals can mange both customer and organizer accounts in addition to managing authorization of Organization accounts

-Organizer: Organizers work as organizer role. Organizers can mange their account. In addition, they can manage their created event

-Customer: customer work as customer role. Customer can manage their account. They can manage their wish list in addition to check-out. Moreover, they can view their history order

-Guest: Guest can register for being a customer account. They can view a wide range of events and detailed events

**2.3.3.2. Document Management System**

**Admin**

|  |  |  |
| --- | --- | --- |
|  | Function name | Description |
|  | Login | Access to system as an admin role |
|  | Logout | Log out of system |
|  | Forgot password | Verify password with OTP code sent |
|  | Update profile | Update information user |
|  | Update avatar | Update avatar for user |
|  | View Customer accounts | List all customer accounts |
|  | View Organizer accounts | List all organizer accounts |
|  | Approve pending organizer | Approve application from guest who want to be a organizer |
|  | Refuse pending organizer | refuse application from guest who want to be a organizer |

**Customer**

|  |  |  |
| --- | --- | --- |
|  | Function name | Description |
| 1 | Login | Access to system as an customer role |
| 2 | Google login | Access to system as customer role |
| 3 | Logout | Log out of system |
| 4 | Forgot password | Verify password with OTP code sent |
| 5 | Change password | Update account with new password |
| 6 | Update profile | Update information user |
| 7 | Update avatar | Update avatar for user |
| 8 | View wish list | Show wish list of user |
| 9 | Add an item to wish list | Add favored event to wish list |
| 10 | Delete an item from wish list | Remove an particular item from wish list |
| 11 | Delete all items from wish list | Remove all items from wish list |
| 12 | View order | View history order by List |
| 13 | Check-out | Check out corresponding to payment method |
| 14 | Search Event | Find event by name |
| 15 | View Events by Category | Show events by particular category |
| 16 | View upcoming events | Show a wide range of upcoming events |
| 17 | View detailed event | Display detailed information by particular event |

**Guest**

|  |  |  |
| --- | --- | --- |
|  | Function name | Description |
| 1 | Register | Register for being an official user. |
| 2 | Search Event | Find event by name |
| 3 | View Events by Category | Show events by particular category |
| 4 | View upcoming events | Show a wide range of upcoming events |
| 5 | View detailed event | Display detailed information by particular event |
| 6 | Register for organizer | Register for being an official organizer. |

**Organizer**

|  |  |  |
| --- | --- | --- |
|  | Function name | Description |
|  | Login | Access to system as Organizer role |
|  | Logout | Log out of system |
|  | Forgot password | Verify password with OTP code sent |
|  | Change password | Update account with new password |
|  | Update organizer profile | Update information organizer |
|  | Update avatar | Update avatar for user |
|  | Create an event | Create an event as Organizer role |
|  | Delete an event | Delete an event as Organizer role |
|  | Update an event | Update an event as Organizer role |
|  | View created event | View event(s) after being created as Organizer role |
|  | View customer list by Order | Show customer list by Order |
|  | View customer list by Account | Show customer list by distinct Account |