Summary:

The customer wants an application similar to Ticketmaster but for users in the fashion industry. Users can manage a guest list, ticket sales, RSVPs, and customer rewards. This application is meant to be user-friendly for non-technical users. It will allow users to make events and stay connected with customers. This application is to help fill a needed gap and relieve creators' stress when hosting an event.

User Stories:

Description	**Status**	**Points**
Error Message of Seating Levels 1. As an event Organizer 2. When I add a seating level with the same category and section already made 3. I should receive an error message.	Completed	3
Booked Seats 1. As an event Organizer 2. When I look at the Seating Summary section 3. Booked Seats should have the number of booked seats from Box Office	Completed	3
Committed Seats 1. As an event organizer 2. When I look at the Seating Summary section 3. Committed seats should be a total of Allocated Seats + Booked Seats	Completed	3
Combine spreadsheet of Manage Invited Guests 1. When I upload a new spreadsheet 2. The current content of the table should not be overridden 3. The table and spreadsheet should be combined	Completed	5
Manage Invited Guests - Empty Email Validation 1. As an event organizer 2. When I upload a new spreadsheet to Invited Guests 3. If the email field has no value 4. I should receive an error message	Completed	2
Manage Invited Guests - Empty Category-Section Validation 1. As an event organizer 2. When I upload a new spreadsheet to Invited Guests	Completed	2

3. If category and section have no value 4. I should receive an error message		
RSVP link 1. As a customer 2. I should receive an email with the RSVP link 3. When I click the link I should be redirected to a webpage 4. I should be able to see how many seats are allocated to me 5. And I should be able to submit my committed seats	Completed	3
Referral link 1. As an event organizer 2. When I send out an email to refer a friend 3. My friend should receive an email to buy tickets with a link 4. My friend should be redirected to the box office	Completed	3
Drop down of event selection 1. As an event organizer 2. When I select the event dropdown 3. I should only see events created by me	Not Completed	2
No drop-down of guest 1. As an event organizer 2. When I select a guest to send an email to 3. The email field should not have a dropdown 4. The guest email should be filled in	Cancelled. Now all emails should be directly sent from the event page.	2
Combined Box Office Sales 1. As an event organizer 2. When I upload a new spreadsheet to Box Office Sales 3. The current spreadsheet should not be overridden 4. The spreadsheets should be combined	Cancelled Now the spreadsheet uploading function acts as a fallback to handle non-public APIs of external box office applications.	5
Box Office Sales - Empty Email Validation 1. As an event organizer 2. When I add a new spreadsheet 3. If the email field has no value 4. I should receive an error message	Cancelled (but may need to be re-implemented.)	2
Box Office Sale - Empty Category-Section Validation 1. As an event organizer 2. When I add a new spreadsheet	Cancelled (but may need to be re-implemented.)	2

3. If category and section have no value 4. I should receive an error message		
Preventing users from committing more seats than the remaining seats. 1. As an event organizer 2. When I add a new guest 3. I should receive an error message when committing more seats than the remaining seats	Completed	3
CRM Authentication 1. As an event organizer 2. I should be able to log in to EventNXT with an Event360 website account. 3. Without logging in and getting an access token, I should not be able to access any service of EventNXT.	Completed	5
Manage Invited Guests - Validation with Seating Levels 1. As an event organizer 2. When I upload a new spreadsheet provided by client 3. If the category and section not match with the category and section of seating level 4. I should get a flag that says “Category and section not found in the seating summary.	Not Completed	3
Eventbrite integration - Connect to Eventbrite 1. When I go to the one of the event pag 2. I should be able to connect my Eventbrite account 3. Then I should see the events I created on Eventbrite	Completed	2
Eventbrite integration -Import ticket sales data from Eventbrite 1. When I go to the one of the event page 2. And I am connected to my Eventbrite account 3. I should be able to select one event from Eventbrite corresponding to the current event page. 4. Then I should see the column names of the Box Office Sales table. 5. 5.Then I should be able to see all the sales of the event from Eventbrite on the Box Office Sales table.	Completed May add improvement to let users map the field names from APIs to the table column names themselves.	4
Eventbrite integration - data validation 1. When I go to the one of the event page 2. And I am connected to my Eventbrite account 3. And I have already mapped the fields and imported the sales data.	Completed	3

4. When some data have conflicts with the Seating Levels table 5. I should see those data being flagged with warnings.		
Manage referral table to be connected with Eventbrite data 1. As an event organizer 2. The table should be managed based on Eventbrite box office data 3. Then the event organizer should be able to see the status of managing referral rewards.	Completed	3
Sent email summary table 1. As an event organizer 2. When I send out an email for RSVP or Referral 3. Email should be sent to the specified email address 4. The table should be updated 5. Email should be updated in this email and removed from the unsent summary table	Completed	3
Unsent email summary table 1. As an event organizer 2. When I send out an email for RSVP or Referral 3. The table should be updated 4. Email should be sent to the specified email 5. Then I will not see the email in this table since it hasn't been sent yet	Completed	3
Event Avatar Upload 1. As a user 2. I want to upload an event avatar in .jpg or .png format, with a file size not exceeding 20 MB 3. So that it displays on the UI with a condensed, web-friendly resolution under 100 KB for optimal performance. 4. Acceptance Criteria: a. The image should display on selection in the UI. b. The resolution should be condensed to under 100 KB.	Completed	3
Event Box Office File Upload 1. As a use 2. I want to upload a Box Office file in .xlsx or .csv format 3. So that the uploaded file is restricted to a maximum of 100 MB.	Completed	3

4. Acceptance Criteria:
    a. Only .xlsx and .csv files should be accepted.
    b. File size limit should be enforced.		

Guest Data Upload Validation with Sample Spreadsheet Download
1. As an event organizer
2. I want to download a sample spreadsheet and receive validation messages when uploading guest data.
3. So that I can ensure that the uploaded file has the correct column headers and required data.
4. Acceptance Criteria
    a. A "Download Sample Spreadsheet" button is present under the UPLOAD GUESTS tab, allowing organizers to download a template with required headers, such as "Name," "Email," and "Section."
    b. If certain required columns, like "Section," have empty values in the uploaded .xlsx or .csv file, display a notification:
        i. Notification Message: "Some columns must contain non-empty data, such as 'Section.' Please ensure these columns are filled before uploading.
    c. The instructional text under the UPLOAD GUESTS tab reads: "The uploaded spreadsheet must have the same column headers as shown in the sample spreadsheet above."
	Completed	4

Referral Reward Deduplication
1. As an event organizer
2. I want the referral reward column to filter duplicate emails.
3. So that each referred email is listed only
4. Acceptance Criteria
    a. No duplicate emails in the referral reward column
    b. Emails shouldn't be displayed side by side if multiple people are referred at once
	Not Completed	3

Sent Email Summary - Duplicate Handling 1. As an event organizer 2. I want the application to alert when sending a referral to an email address that has already received the referral from the same guest.	Not Completed	3

3. So that I can prevent duplicate referrals while allowing the option to resend from the sent email list.
4. Acceptance Criteria
    a. Duplicate email alerts in the sent email summary
    b. Option to resend from the email list		

RSVP Invitation HTML Removal 1. As an event organizer 2. I want to prevent HTML elements from displaying in RSVP invitations. 3. So that the invitation looks clean and user-friendly 4. Acceptance Criteria a. No HTML elements in the RSVP invitation.	Completed Partially (Removed that part from the Email-Service Page from every table, it will not be shown unless the user chooses to edit the template itself.)	3

Email Template Management
1. As an event organize
2. I want the email template options hidden by default
3. So that I only see the "MODIFY Template" button until clicking to view editing options.
4. Acceptance Criteria
    a. "MODIFY Template" button is visible
    b. Template editing options (EDIT/DELETE/ADD) appear upon clicking.
	Completed	3

Ticket Type Reporting Parsing 1. As an event organizer 2. I want the app to parse ticket types accurately by separating Category and Section into distinct fields. 3. So that reports display organized data without requiring manual edits. 4. Acceptance Criteria a. Category and Section parsed correctly	Completed	3
Email Service Bulk Send 1. As an event organizer 2. I want to send multiple emails simultaneously via the email service 3. So that I can efficiently reach multiple contacts 4. Acceptance Criteria a. Multiple emails are to be sent in one batch	Completed partially	3
Rotten Tomatoes & Tickettomato.com API Integration 1. As an event organizer	Cancelled due to no public API support. As a fallback,	4

2. I want to integrate with Rotten Tomatoes and Tickettomato APIs within the app 3. So that event-related information can be enriched with external data. 4. Acceptance Criteria a. Successful integration and retrieval of API data	re-implement the spreadsheet uploading function.	

Understanding legacy code:

Our strategy for learning and improving the code was to go through the code from the previous group, deploy the application to see how it works so far, and communicate with the client to see what needs were and still need to be met. We worked together as a team communicating through Slack to set up meetings to meet everyone's schedule and to help each other when problems occur. A linear account was made for our group to assign work, see what everyone is working on, and see what has been completed so far.

The previous group's GitHub repository listed their emails and detailed instructions on how to run the application. Notes were included in the final report on how we should proceed from where they left off. After running the application like the previous team instructed, we found multiple bugs needing attention before continuing to add features. In sprint 1 we decided to fix all the bugs found so no fixes and new features were added. From sprint 2 to 4 we added new features to meet the client's needs.

Sprints Roles:

1. **Sprint 1**
    a. **Project Manager - Rishabh**
    b. **Scrum Master - Kiara**
    c. **Developers - Ming, Surya, Neil, Parth**
2. **Sprint 2**
    a. **Project Manager - Kiara**
    b. **Scrum Master - Ming**
    c. **Developers - Rishabh, Surya, Neil, Parth**
3. **Sprint 3**
    a. **Project Manager - Neil**
    b. **Scrum Master - Rishabh**
    c. **Developers - Surya, Kiara, Parth, Ming**
4. **Sprint 4**
    a. **Project Manager - Surya**

    b. **Scrum Master - Parth**
    c. **Developers - Neil, Kiara, Ming, Rishabh**

Scrum Iterations:

Sprint 1

The team worked on going through the legacy code and solving the bugs to fix the broken functionalities of the application. Sprint 1 achievements are as follows:

1. Environment set up
    a. Running the legacy application locally
    b. Testing out the legacy application locally
    c. Deploying the application to the platform Heroku
    d. https://linear.app/eventnxt/issue/EVE-16/manually-deployment
2. The team worked on a few bugs and fixed the functionality for the same.
    a. Users cannot upload a guest list file.
    b. The email service is not working.
    c. The sign-out feature is not working as expected.
    d. Manage seating level

Sprint 2

The team completed the majority of the user stories for the improvements and rspec fixes are completed. Sprint 2 achievements are as follows:

- Seating level and Invited Guests are fixed
- Login and logout feature is fixed
- Overall progress with the application is being made.

Sprint 3

The team completed the majority of the user stories for the improvements and rspec fixes are completed. Sprint 3 achievements are as follows:

- Connecting the app through the Eventbrite external API for box office data.
- Login and logout feature implemented using the main events 360 CRM tool
- Email services like referral rewards and RSVP invitations have been enhanced.

Sprint 4

The team completed most user stories and showed the client all the features we worked on. Sprint 4 achievements are as follows:

- Added a download option under Manage Invited Guests where the event owner can download the entire guest list.
- Added a hover option where the event owner can see the requirements before uploading the sheet
- Hidden email template summary. The event owner can see the data only when he clicks the modify button.
- Modified the event avatar upload requirements. The event owner can upload only .png/.jpg with a maximum size limit of 20 MB.

- Moved the email dashboard under the event dashboard where the event owner can send emails directly under the event dashboard. This was not completed fully and as per discussion with the client, this will reduce the burden on the upcoming team who is going to work on this. We made a separate branch and document explaining this feature so that the next team could work without any issues.
- Made sure no robocop issues and code coverage above 90%

Sprints Contributions

	Neil	Kiara	Surya	Parth	Ming	Rishabh
	User Stories | Points					
Sprint 1	Bug Fixes	Bug Fixes	Bug Fixes	Bug fixes	Bug fixes	Bug Fixes
Sprint 2	3 | 9	0 | 0	1 | 5	2 | 4	0 | 0	3 | 9
Sprint 3	0 | 0	3 | 9	3 | 9	3 | 9	3 | 9	0 | 0
Sprint 4	3 | 9	2 | 6	0 | 0	0 | 0	3 | 9	3 | 9
Total	6|18	5| 15	4| 14	5| 13	6 | 18	6 | 18

Customer Meetings

Sprint 1 MVP Demo
Date: 9 October, 2024
Time: 1:00 p.m.
Place: Zoom

Feedback
The client was not too happy with the team not being able to complete new features. The team explained to the client about the bugs needing to be fixed before adding new features and/or improvements to the application. Although the client still was not happy with the fixes the team made to the application, the team was able to receive more user stories from the client for the next sprint.

Sprint 2 MVP Demo
Date: 23 October, 2024
Time:
Place: Zoom

Feedback

The customer was satisfied with the amount of work the team has completed so far. The customer still had a couple of concerns we are addressing. The customer also gave feedback to improve the features more to make it more user-friendly.

Sprint 3 MVP Demo
Date: 6 November, 2024
Time:
Place: Zoom

Feedback
The customer was satisfied with the amount of work the team has completed so far. The customer still had a couple of concerns we are addressing. It was mainly decided to host everything into 1 branch and share the link with the client to test and give a sign-off.

Sprint 4 MVP Demo
Date: 20 November, 2024
Time:
Place: Zoom

Feedback
The client was happy with the work we had done. We showed him all the features we completed till sprint 4 and explained everything. He tested everything and was working without any issues. Also shared the deployed Heroku app.

BDD/TDD Process

Applying BDD/TDD effectively in the development process can be challenging. Several issues we encountered have made this process more difficult. First, the legacy code is buggy, and the tests written for it are not passing, despite 90% test coverage. This makes it harder to understand the fundamental requirements of the project, forcing us to spend additional time fixing these issues based on our limited understanding.

Second, the original implementation of the project included features that did not meet the client's requirements, and these discrepancies were not documented in their report. This has led to high communication overhead between our team and the client, as we struggle to understand the incorrect features. Consequently, we have difficulty determining the right direction for improving or re-implementing them and producing accurate user stories.

Third, as the product owner role rotates between team members, and given our inexperience in taking on this responsibility, we face challenges in maintaining continuity and ensuring a clear vision for the product. This adds complexity to the decision-making process and creates inconsistencies in the overall direction.

Eventually, due to the user stories we created having misaligned priorities or incorrect directions, a significant amount of time was spent re-writing tests. These revisions often felt redundant, further delaying progress as we continually adjusted the tests to reflect updated requirements and priorities.

Configuration Management:

Our configuration management approach involves using the `dev` branch as the primary branch for development and testing. We ensure that the `dev` branch is fully functional on Heroku, our production environment. After each sprint, we merge the `dev` branch into the `main` branch, which serves as the production-ready version.

For continuous integration (CI), we utilize GitHub Actions to run automated checks for RSpec, Cucumber, and RuboCop. When developing a new feature, team members create a feature branch from the `dev` branch and submit a pull request (PR) on GitHub. Before merging the PR into the `dev` branch, it must be reviewed by at least one team member—typically the product owner or scrum master—and pass all CI checks to ensure high test coverage and code quality. This process guarantees that only thoroughly tested and reviewed code is merged into the main development branch.

Additionally, we use Docker to ensure that all developers' environments are consistent and closely resemble the production environment. This helps eliminate environment-specific issues and streamlines the development process across different team members.

Tools: Github/Git, Docker, SimpleCov, Rubocorp, Rspec, Cucumber**.**

Repo Contents:

The deployment and testing scripts are located in the `scripts/` folder, and the steps for usage are outlined in the `README.md` file.

The code inherited from the previous team was badly disorganized, and due to time constraints, we were unable to resolve all of the issues. We focused on removing redundant files, code, and comments from the legacy project, cleaning up configuration files, and using RuboCop to address code complexity. However, there are still areas that have not been fully addressed, such as over-complex view files.

GitHub link: FashioNXT/EventNXT-Fall2024

Heroku link: https://eventnxt-fall2024-main-b1b544302697.herokuapp.com/

Presentation and Demo video link: EventNXT final project demo