

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TEXAS A&M UNIVERSITY, COLLEGE STATION



CSCE 606: SOFTWARE ENGINEERING

CRMNXT FINAL REPORT

Team Members

Amith Kini Mattar

Prachi Miyan

Jason Li

Samraat Gupta

Yen Hua Lin

Rishab Nair (Q-Dropped)

Scrum Master & Project Manager (On rotation)

Iteration0: Rishab Nair, Jason Li

Iteration1: Jason Li, Prachi Miyan

Iteration2: Amith Kini Mattar, Prachi Miyan

Iteration3: Prachi Miyan, Samraat Gupta

Iteration4: Yen Hua Lin, Rishab Nair

Iteration5: Samraat Gupta, Yen Hua Lin

Summary:

CRMNXT is a comprehensive CRM app for FashionNXT which comprises EventNXT, PlanNXT, CastNXT and NXTFolio. Events360 is the one-stop-shop to access all these applications. Our customer highlighted that he wanted a unified user dashboard implemented along with a unified login process for all NXT applications via CRMNXT. He had also specified the need for graphs to view user analytics as well as the ability to send emails to specific user types of each app. He wanted us to implement functionality to add new NXT apps to the CRMNXT suite for user authorization.

Throughout the semester we implemented functionality to meet these requirements. A user can be registered as a “user” type or an “admin” type within CRMNXT. While a “user” type can login to any of the member apps through OAuth via CRMNXT, an “admin” type can login to CRMNXT to access user data for each of the apps to utilize user management, analytics, and email functionality. User analytics pages for each app display graphs by processing user specific data. Meanwhile, the email feature allows an admin to send emails to individual users and user types for each app as well as save email templates for future use. We also streamlined the process to add new apps to the dashboard and authorize their users to login to the apps through CRMNXT. The stakeholders for CRMNXT are the FashionNXT employees that use this app to enhance management of the NXT app suite as well as all the users of the NXT app suite. CRMNXT enhances user experience for the entire FashionNXT portfolio and allows the admins to access insights pertinent to enhancing management of their apps.

User Stories:

Project Setup / Refactoring:

User Story	Person Assigned	Story Points	Status	Description
Application deployment and minor code cleanup	AM	1	Done	
Restart Email Service	SA	2	Done	Create an email address so that emails can be sent and received for the application. This is not a long term solution, but a band-aid on the existing problem.
CSS Clean Up	JA	2	Done	
Refactor Javascript from all the pages	AM	2	Done	Chore: Javascript is scattered all over the HTML files, cluttering the application templates and inefficient implementation. This task removes the Webpack framework, and cleans up the javascript code.
Install and enable selenium	PM	2	Done	
Add Continuous Integration with GitHub actions	AM	1	Done	Add GitHub actions to automatically test incoming PR and merges.
Setup simpleCov metric	JA	1	Done	
Setup OAuth provider and come up with API endpoints required from other application	AM	2	Done	As a user I want to login to NXT application, via Events360 So that I can access the NXT application
Transfer Heroku application to the Client account	AM	1	Done	Refer Canvas Inbox: "NXT Heroku Account"
Import Chartjs library	JA	2	Done	
Import Bootstrap	JA	2	Done	

New Features:

User Story	Person Assigned	Story Points	Status	Description
Allow customizable content in posts- attach image to post	PM	1	Done	As an admin of the CRM service I want to have the option to attach images to the posts that are created and view them when they are saved So I can customize the posts that I create
Forgot password page	SA	3	Done	As a user I want to reset my password So that I can login with a newly set password
Modularize adding new OAuth application to the system	AM	3	Done	As an admin user I want to create a new NxtApp So that users of that NxtApp can use my platform for OAuth authentication
Updates login validation	JA	2	Done	
Mailer when a user signs up for an admin account	SA	3	Done	As an admin I want to be able to accept/deny request for admin access
Create email templates and fix not working links	YE	3	Done	As a user I want to use template when sending emails So that I don't need to type a word When I select a specific template And click fill in Then the title and content of the selected template will automatically fill in the corresponding blank
Add foreign key to email templates and posts	YE	3	Done	
Email Recipient and fix not working links	YE	3	Done	
Setup Password Recovery	N/A		Iceboxed	
Analytic of each service	N/A		Iceboxed	As an admin of the services I want to see analytic information from users of the website So I can be informed about the preference of my customers
Separate User model and Admin model for security purposes	N/A		Iceboxed	Separate the user model and admin model, so that only user models can use OAuth, and admin models can only login to the dashboard.

UI/UX changes:

User Story	Person Assigned	Story Points	Status	Description
Toggle between services in user management	YE, AM	2	Done	As a CRM customer I want to toggle between different services in FashionNXT So that I can access information about different services that are available
Home & Logout button	JA	1	Done	As a user of the service I want to have the link to the home page So I can navigate to the home page from any page of the website.
Create CRMNXT page (New dashboard)	JA	1	Done	As a CRM user I want have the option to select between PlanNXT, CastNXT, and EventNXT So I can navigate to the CRM interface for each application
Router rework & dynamic page title	JA	1	Done	As a CRM user I want to see the app type on the title of the page So I know which application I am on right now
User type radio button for registration page	YE	2	Done	As a user/admin, I want to select my user type on registration page So that I can create my account with correct accessibility
Checkbox for user permission	YE	1	Done	As an admin, I want to manually decide the permission of a user So that I can denied the access for those users who doesn't pay
Create HTML admin page that accesses data from required API, parses the data, and populates the users management page. write cucumber tests	PM	2	Done	Assess users from the different NXT apps using APIs.
Display 3 graphs in Analytics page for user type, user status and last access date for each separate app -	PM	3	Done	Each analytics page displays access to its specific API (using mock APIs that are also being used by user management feature), filters the data in the analytics_helper and displays the

Iteration 4				relevant information in the form of graphs.
Setup user dashboard page	AM	2	Done	When the user logs in, they need to land on a different page than the admin. This landing page should direct them to the app pages.
Add histogram in user analytics to show the distribution of fee collected from the users in the last 30 days	PM	1	Done	
Add selectors for user analytics graphs. Add CSS to display the graphs neatly	PM	2	Done	
CRMNEXT Login Page	SA	3	Done	
UI Rework	JA	3	Done	
Design Logos for the three NXT apps and CRMNEXT page	SA	3	Done	As per request of the client, the homepage should show the logos of the apps in the NXT suite. The logos have specific font requirements.
Add more analytics to the top of the analytics page for each app	PM	2	Done	

Researching/Communications:

User Story	Person Assigned	Story Points	Status	Description
Reach Out to the Other Teams and deploy for analytics	RM	2	Done	
Help out other teams with OAuth and API integration	AM, PM	2	Done	
Investigate Active Analytics	N/A		Iceboxed	

Understanding the existing code:

We made several modifications to the legacy code since it did not align with the customer's vision. We found that the code had a broken email system and the previous team often hard coded functionality. We cleaned the css, refactored the javascript from all the pages, and restarted the email service system. We also found most of the legacy tests failed so we refactored all of the tests. User analytics that were displayed in the analytics page tracked visits to the CRM application itself which was not very useful for the client. We changed that to display graphs for analytics of each of the applications separately. We also removed the companies button in the legacy app which did not serve any useful purpose for the application. We built upon the legacy code to enhance the features to align with the client's vision.

Iterations:

Iteration 0:

- Organized the team and developed a meeting schedule.
- Connected with the client to understand the project requirements and goals.
- Listed down user stories and submitted a design document.
- Acquainted ourselves with the technologies being used.

Iteration 1:

- In this iteration, we strived to learn how our CRM Application works, as it is a legacy application.
- We successfully [deployed our application on Heroku](#).
- The previous team left us a broken email module. By investigating into the issue, we successfully [restarted the email services](#).
- At the same time, we started with some simple UI changes such as [Toggle between services in user management](#) and [Adding home and logout buttons](#).
- By [establishing connection with the other teams](#), we paved the way for future integration.
- We communicated and worked together with our client, outlining future features to be implemented.

Iteration 2:

- We noticed that many codes are poorly organized and overly repetitive in our application. Therefore, we focused on refactoring the CRM modules this iteration, including [CSS Cleanup](#) and [Refactoring Javascript from all the pages](#).
- To fulfill client requirements, we implemented [the new Event 360 dashboard page](#) in addition to CRM services.
- As an essential part of user management service, the [forgot password workflow](#) was added to the system.
- After great effort, we [integrated Selenium testing into our application](#). This gives us the ability to test JavaScript modules.

Iteration 3:

- [Github actions - CI](#) was implemented to automate the testing process. In addition, [simpleCov metric is set up](#) to display code coverage.
- We created [an application management system](#), which paved the way for future OAuth integration.
- More UI changes are implemented along the way: [type radio button for registration page](#), [dynamic page title](#), and [checkbox for user permission](#).

Iteration 4:

- The [core utility of the OAuth module](#) is implemented, which enables modularized integration of OAuth application to the system. From this point on, our application accepts remote API data from the other NXT teams.
- We focused on CRM analytics in this iteration. After [ChartJS setup](#), [graphs for user type, user status and last access date](#) is implemented.
- The email module was reworked to [allow templates for posts](#).
- We added another layer of user access control: [admin account registration now requires the owner's email approval](#). [User validation logic](#) is also refactored.

Iteration 5:

- Most UI beautification takes place in this iteration. With the [help of Bootstrap5](#), we were able to [rework the entire frontend UI](#). Our application now has a view that looks closer to real CRM tools in the market.
- As per the client specifications, we added a new [homepage](#) for the Events 360 application suite which shows the different apps in the suite and displays information about them. The user can go to the login page from this page.
- For the user analytics side we added a new [histogram](#) that shows the distribution of fees for the last 30 days and added [toggles](#) so that the user can check which graphs should be displayed.
- For the email functionality of our app, we added the option to select the [recipients](#) by user types so we can easily send emails to users of a particular type such as talent or producer etc.
- An integral part of the CRM app is [integration](#) with other apps so we created documentation as well as a mock application to help the other teams integrate our CRM services into their apps particularly for OAuth and data analytics.

Customer Interaction:

We had a zoom meeting with our client Mr. Tito Chowdhury on following days:

2/7/2023 at 1.00 pm CST on Zoom
2/16/2023 at 1 PM CST on Zoom
3/2/2023 at 1 PM CST on Zoom
3/16/2023, 3/30/2023 at 1 PM CST on Zoom
4/13/2023 at 1 PM CDT on Zoom
4/28/2023 at 11:50 AM CDT on Zoom
5/5/2023 at 15 PM CDT on Zoom

We had zoom meetings with our client Mr. Tito Chowdhury on the days above for every Iteration.

- Customer meeting 1: Connected with our client on 7th February in Iteration 0 to understand the project requirements and goals.
- Customer meeting 2: Connected with our customer on 16th February in Iteration 1 to clarify user stories that needed to be implemented to build CRM Service.
- Customer meeting 3: Connected with the customer on 2nd March before Iteration 2, in which we discussed our progress, clarified the customer needs. He asked us to fetch data from other applications.
- Customer meeting 4: We connected with the client on 16th March in Iteration 3 and demonstrated the changes, took some feedback, and discussed the future stories. He asked us to connect to other teams to integrate OAuth.
- Customer meeting 5: We connected with the client on 13th April in Iteration 4 to demonstrate the changes which we implemented till then (heroku application transfer, login validations, graphs in analytic pages, admin request emails, and email templates). He asked for other applications to setup API connections to us.
- Customer meeting 6: We connected with the client on 28th April in Iteration 5 to demonstrate the core CRM features, OAuth, sending emails and analytics features which we implemented till then.
- Customer meeting 7: We connected with Mr. Tito Chowdhary on May 5th to show our final application.

Design:

CRMNXT Home Page:

Events 360

Login

<

CASTNXT

>

Events 360 is a A wrapper web-app that contains three web-Apps that automate three major aspects of any events planning.

CASTNXT

Talent casting App for staging performance (e.g. fashion show, on-stage drama performance) automation.

View details »

EVENTNXT

Web-app to automate the tracking of details in an event setup plan.

View details »

PLANNXT

Event Guest List Automation - Live events commit tickets through two channels -- (1) Box-office ticket sales and (2) Invited Guest list RSVPs.

View details »

NXTFolio

Creative Industry visual "LinkedIn", matching App for Creative/Visual/Fashion industry professionals more accurately than existing apps in the market.

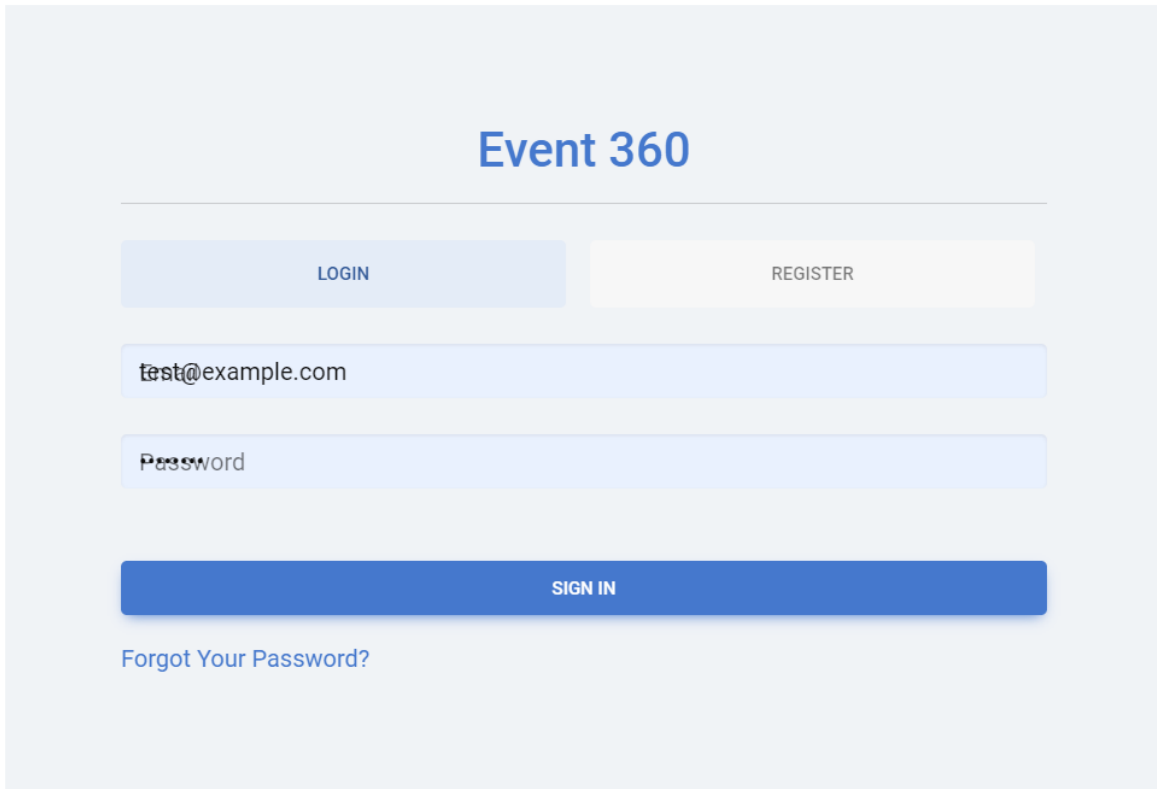
Learn more

NXTFOLIO

CREATE. CONNECT. CREATE.

9

Login Page:



The login page for Event 360 features a light blue background. At the top center, the text "Event 360" is displayed in a bold, dark blue font. Below this, there are two buttons: a light blue "LOGIN" button and a light yellow "REGISTER" button. Underneath these buttons are two input fields: the first contains the email "test@example.com" and the second contains the password "Password". Below the input fields is a large, solid blue "SIGN IN" button. At the bottom left of the form area, there is a link that says "Forgot Your Password?" in a smaller, dark blue font.

Event 360

LOGIN REGISTER

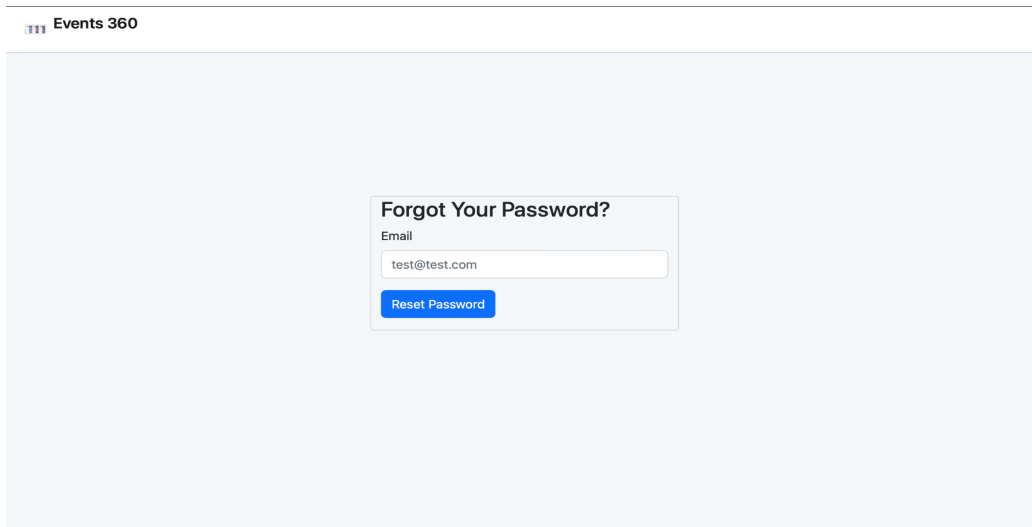
test@example.com

Password

SIGN IN

[Forgot Your Password?](#)

Forgot Password Page:



The forgot password page has a light blue background. At the top left, there is a header with a small icon and the text "Events 360". In the center of the page, there is a white box with a light blue border. Inside this box, the text "Forgot Your Password?" is at the top. Below it, the word "Email" is followed by an input field containing "test@test.com". At the bottom of the box is a blue button with the text "Reset Password".

Events 360

Forgot Your Password?

Email

test@test.com

Reset Password

Registration page:

Event 360

LOGIN

REGISTER

User Type

☒ User ☐ Administrator

Name

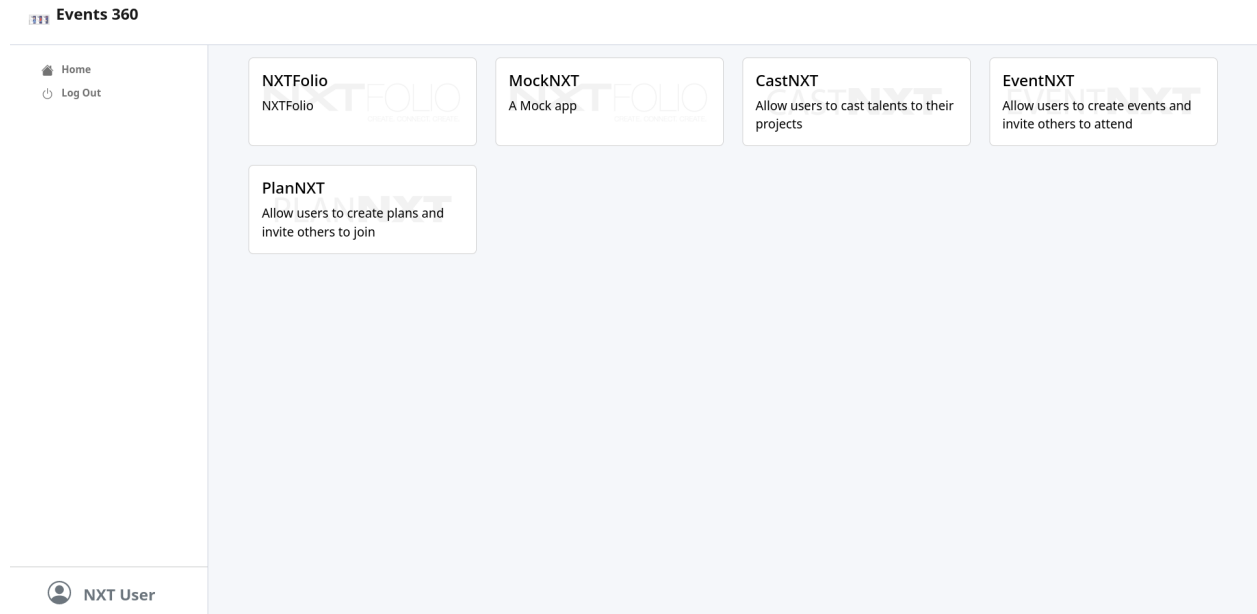
Email

Password

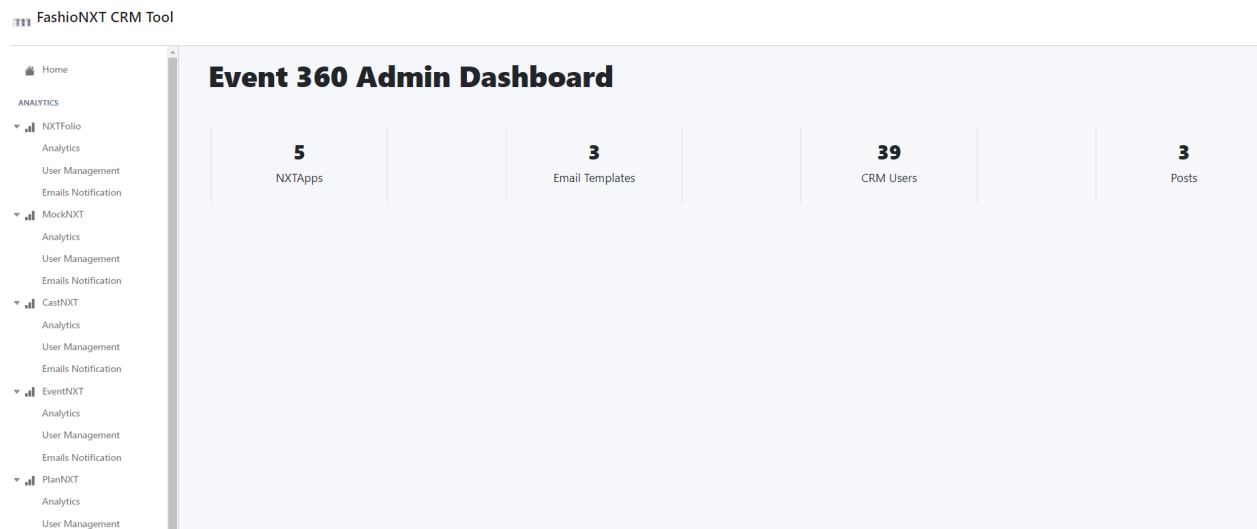
Re-enter password

REGISTER

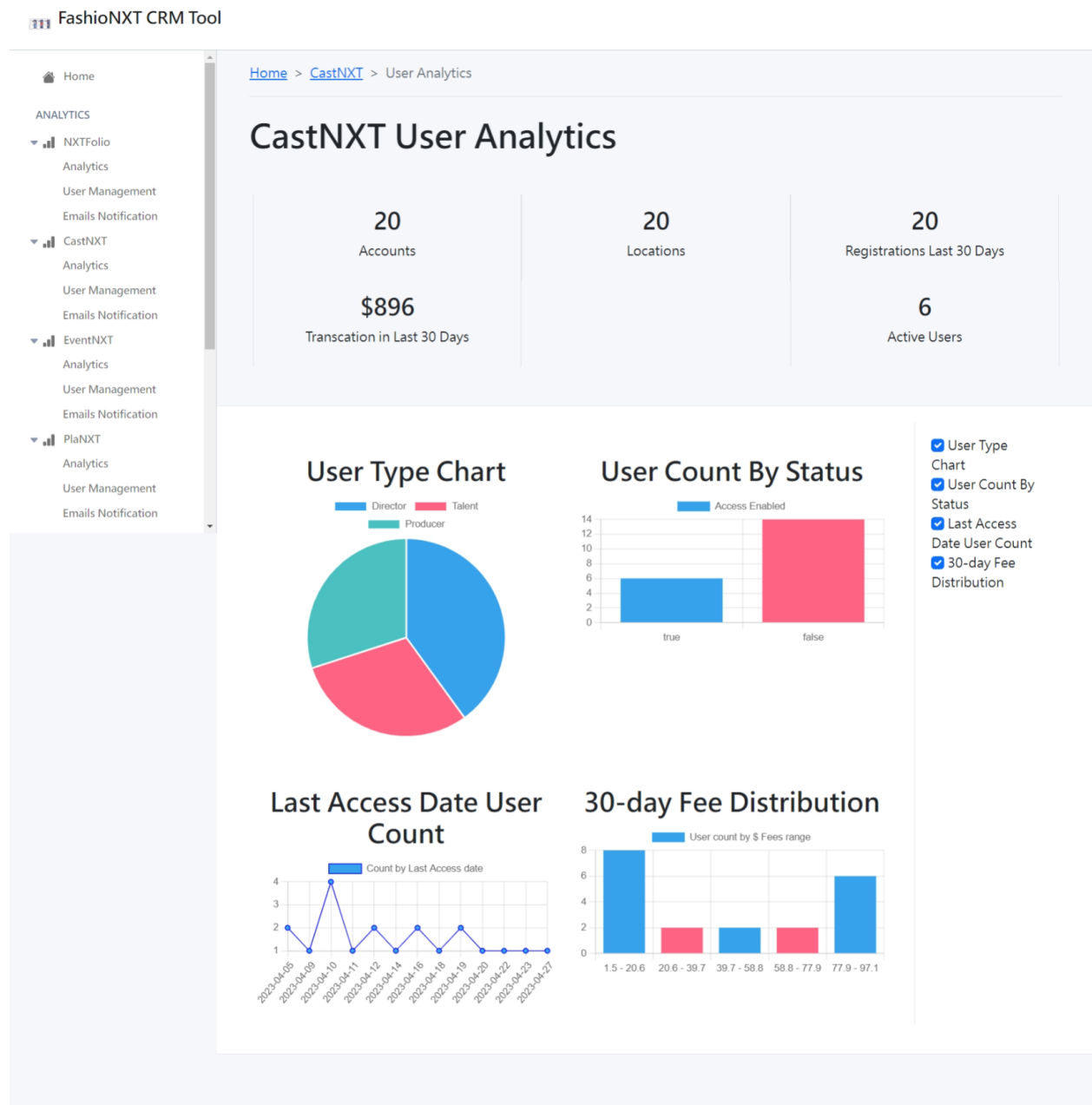
User Dashboard:



Admin Dashboard:



User Analytics:



User Management:

Home

ANALYTICS

NXTFolio

Analytics

User Management

Emails Notification

CastNXT

Analytics

User Management

Emails Notification

EventNXT

Analytics

User Management

Emails Notification

PlaNXT

Analytics

User Management

Emails Notification

Home > CastNXT > User Management

CastNXT User Management

User ID	Name	Email	Membership Type	Location	Created at	Avg minutes last 30 days	Fees from transaction last 30 days	Access Enabled
1	Christina Hampton	stevennunez@example.org	Director	Myersmouth, VI	2023-05-05T16:19:00	33	52.07	true
2	Kendra Brewer	laurahayden@example.com	Director	North Aprilfurt, MS	2023-05-05T16:19:00	26	1.48	false
3	Jessica Ward	rogerseric@example.net	Talent	Stephanieville, VA	2023-05-05T16:19:00	32	89.68	false
4	Andrea Thomas	perrybush@example.com	Producer	Lake Michaelhaven, MA	2023-05-05T16:19:00	30	15.9	false
5	Adam Avila	willisdana@example.org	Talent	Scottfort, TN	2023-05-05T16:19:00			
6	Elizabeth Carter	jacksondavid@example.com	Talent	Port Miaton, MA	2023-05-05T16:19:00	24	20.17	true
7	Marcia Vincent	melindavega@example.org	Producer	Jensenborough, NY	2023-05-05T16:19:00	30	97.05	false
8	Tara Reed	kathrynhaley@example.com	Director	South Kelly, AZ	2023-05-05T16:19:00	28	5.73	false
9	Erin Deleon	april47@example.net	Producer	Hardyburgh, IA	2023-05-05T16:19:00	32	83.43	false
10	Richard Peterson	cmitchell@example.net	Producer	Paulatown, VA	2023-05-05T16:19:00	33	13.69	false
11	Seth Taylor	mathew54@example.org	Talent	East Lisatown, AK	2023-05-05T16:19:00	30	82.15	false
12	Shannon Reed	andrewjohnson@example.org	Talent	Austinhaven, PA	2023-05-05T16:19:00	26	72.15	true
13	Zachary Jordan	josephjohnston@example.org	Director	Alejandrofurt, DC	2023-05-05T16:19:00	34	8.18	false
14	Holly Sanders	lryan@example.org	Director	New John, VT	2023-05-05T16:19:00	31	57.58	true
15	Seth Atkins	michele24@example.org	Talent	Victoriashire, KY	2023-05-05T16:19:00	29	3.17	false
16	Benjamin Hernandez	angelasullivan@example.net	Director	Richardsonshire, AZ	2023-05-05T16:19:00	33	26.89	false
17	Kayla Wheeler	holly58@example.com	Producer	Humphreyview, AL	2023-05-05T16:19:00	34	7.37	true
18	Kelli Davis	wrightann@example.com	Director	Port Vanessa, CA	2023-05-05T16:19:00	32	89.62	true
19	Valerie Phillips	walshcorey@example.org	Producer	Hollandchester, GA	2023-05-05T16:19:00	30	62.06	false
20	Andrew Alvarez	gordonthomas@example.net	Director	Lake Taylor, MD	2023-05-05T16:19:00	30	82.25	false

Back to User Analytics

Email Notifications:

Home

ANALYTICS

NXTFolio

Analytics

User Management

Emails Notification

CastNXT

Analytics

User Management

Emails Notification

EventNXT

Analytics

User Management

Emails Notification

PlaNXT

Analytics

User Management

Emails Notification

Home > CastNXT > Email Notifications

CastNXT Email Notifications

ID	Title	Content				
6	Test	Test	PDF	Show	Edit	Destroy

Import Reports

[New Report](#)[Templates list](#)[Reports PDF](#)

New Post:

Home

ANALYTICS

NXTFolio

Analytics

User Management

Emails Notification

MockNXT

Analytics

User Management

Emails Notification

CastNXT

Analytics

User Management

Emails Notification

EventNXT

Analytics

User Management

Emails Notification

PlanNXT

Analytics

User Management

Emails Notification

MANAGEMENT

Application Management

CRM Users

Log Out

Home > PlanNXT > Email Notifications > New Post

New Post

Recipient

Title

Content

Image

Browse...


No file selected.

Send Post

Recipient types: [Vendor](#) [Client](#) [Staff](#) [Planner](#)
[Templates list](#)
[Back](#)

15

Email Templates:

 FashionNXT CRM Tool

Home > [CastNXT](#) > [Email Notifications](#) > Email Templates

CastNXT Email Templates

[New Template](#)

Title	Content	
Test	Test	Fill In

[Back](#)

Testing:

We developed cucumber test cases for the BDD processes (located in features/ directory). We also wrote integration test cases for each controller and helper classes that we made. Rspec test cases are used for unit testing (located in spec/ directory). In addition, Selenium testing is used to test Javascript modules (located in test/ directory).

- Run the RSpec tests using the command: `rspec`
- Run the Cucumber tests using the command: `cucumber`
- The above mentioned tests can also be run together using the command: `rake`
- Selenium tests can be run using the command: `rake test:system`

Since Github CI workflow is implemented, the above tests are executed automatically on every pull request.

The following is the output of RSpec and Cucumber testing and the coverage report:

```
rspec
.....

Finished in 0.52622 seconds (files took 2.58 seconds to load)
48 examples, 0 failures

cucumber

45 scenarios (45 passed)
219 steps (219 passed)
0m49.187s















Coverage report generated for Cucumber Features, RSpec to
/home/amithmkini/Assignments/Software_Engineering/FashionXT-CRM-Servi
ce/coverage. 401 / 479 LOC (83.72%) covered.
```

The total coverage achieved was 83.72%. We increased the coverage by removing unused code, and implementing tests for legacy features.

Repo Contents

For every new iteration each member creates a new branch from master. After completing all the changes and testing locally, each raises a PR to the master. At the end of each iteration, we merge all branches created to master and deploy it as the “production branch”. We created a new tag for every release at the end of each iteration with the “IterationNo.” and associated it with the releases as well.

Tags		
Iteration5	<div> <div>last week</div> <div>2698f29</div> <div>zip</div> <div>tar.gz</div> </div>	<div>Verified</div> <div>...</div>
Iteration4	<div> <div>3 weeks ago</div> <div>6a9c9cd</div> <div>zip</div> <div>tar.gz</div> </div>	...
Iteration3	<div> <div>on Mar 31</div> <div>f39e3a7</div> <div>zip</div> <div>tar.gz</div> </div>	...
Iteration2	<div> <div>on Mar 10</div> <div>da70369</div> <div>zip</div> <div>tar.gz</div> </div>	...
Iteration1	<div> <div>on Feb 24</div> <div>82b8e74</div> <div>zip</div> <div>tar.gz</div> </div>	...

Default branch		
	<div>master</div>  Updated 30 minutes ago by xup6u04cj86	
Your branches		
	<div>email_receiver</div>  Updated last week by amithmkini	
Active branches		
	<div>HomepageUpdate</div>  Updated 36 minutes ago by stingyemperor	
	<div>null_check</div>  Updated 1 hour ago by pmiyan	
	<div>minorFixes</div>  Updated 2 days ago by amithmkini	
	<div>homeBootstrap</div>  Updated 3 days ago by amithmkini	
	<div>test_correction</div>  Updated 3 days ago by pmiyan	
View more active branches		

Local Setup:

1. Clone the GitHub repo: `git clone git@github.com:tamu-edu-students/FashionNXT-CRM-Service.git`
2. `cd FashionNXT-CRM-Service`
3. Install bundler for Ruby: `gem install bundler`
4. Install the app requirements: `bundle install`
5. Setup the local sqlite DB: `rake db:create db:migrate`
6. Set ENV vars for mailer by creating a file `config/local_env.yml` (copy the format from `config/local_env.yml.example`)
7. Run the app using: `rails s`
8. Visit `http://127.0.0.1:3000` to see the Application running.

Heroku Deployment:

The deployment has been done using the customer's account, redeployment would not be necessary. The following team should ask the customer for access to the account.

For Heroku deployment before, we follow the steps below:

1. [Deployment guide for rails application](#)
2. Install [Heroku CLI](#)
3. Login to Heroku using the command: `heroku login`
4. Create a Heroku app using the command: `heroku create`
5. Push the changes to Heroku using the command: `git push heroku main`
6. Run db migrations: `heroku run rake db:migrate`
7. Set ENV vars according to `config/local_env.yml.example` in Heroku dashboard

Issues faced when releasing product to Heroku:

1. The initial deployment failed due to Procfile not being configured to run migrations before deployment. This was fixed by editing the Procfile to include the same.
2. On subsequent deployments, we relied on GitHub master branch commit trigger to auto-deploy on Heroku. One shortcoming of this was the possibility of broken production state. We fixed this issue with GitHub CI which ran the tests and Heroku waited for these tests to pass before deploying the code.
3. Sometimes, due to improper migrations, the defaults were not propagated to the existing values, leading to errors in the production. We had to manually connect to the Postgres DB to fix the missing fields and get the app back up.

Issues / Challenges

1. Since a part of our app requires collaborating with multiple teams, we had to reach out to the other teams in order to integrate OAuth and data analytics. Since communication between the teams was mostly done through email, it was slow and tedious.
2. The legacy code contains many modules that were malfunctioning. We needed to restart the email services before getting our application back to work. Many of the views and controllers were hardcoded, repetitive and poorly formatted, which required extra refactoring in later iterations. The application was undeployed by the previous team, which required manual deployment.
3. The client had many vague expectations for our project and a lot of the features were left for us to figure out while at the same time the client demanded new features in each meeting. This led to us having to change the direction of the project multiple times during the semester.
4. Near the end of the semester, since the integration with other NXT apps was taking too long to be implemented, we had to set up our own mock app in order to show the functionality for our CRM. This was something that we had not thought of before and something we had to work on last minute.
5. EC2 connection dropped often and had limited storage.

Gems Used

1. **Wicked_pdf** : this was used for the pdf downloader. This converts the html view created by us as a pdf.
2. **Rspec, rspec-rails, cucumber-rails, selenium-webdrivers, rail-controller-testing** : All these gems are added for testing purposes. They are used in the TDD/BDD process.
3. **Ahoy_matey**: This gem is used to track events and visits and the data is automatically stored to the local database. This was very helpful for the analytics part where we needed to track the users data. Although this sits only in the backend at the moment it can be used in the CastNXT and EventNXT applications and results can be published in the CRM.
4. **Active_analytics**: This gem helped generate the view count for the application and the pages within it.
5. **Net-smtp, net-imap, net-pop**: These are used for our email services to provide functionality to send mails via SMTP calls. At present these are not used as we are sending mails over the server.
6. **Doorkeeper**: This gem was used to create and host the OAuth Provider.
7. **Carrierwave**: This gem was used for attachments in emails.
8. **Httparty**: This gem provides an interface for making HTTP requests to APIs.
9. **Simplecov**: This gem provides code coverage analysis.

Software Development Process:

Our team followed BDD and TDD to develop features for the app. It helped us map the customer's requirements to tests that could then be implemented to pass the tests. This helped streamline the development process by not only speeding up development process but also avoiding breaking old features while we developed new ones. Github CI helped us make sure that the new changes did not break any features on our application.

Utilizing pivotal tracker helped us with implementing agile methodology of development. Our team interacted often in person and over zoom meetings to ship new functionality often.

Future Scope

- The team could implement APIs for CRMNXT which will provide other applications with the ability to directly store the information in CRMNXT's database instead of CRMNXT using the application's API to pull in data. This will make integration of new apps more streamlined and modular.
- The team can also add more graphs to the analytics section, like one to monitor the time that users have been active on a web application. This will help administrators with CRM.
- According to the client specifications, the team can also integrate active analytics with the other apps so that data such as number of current active users,time spent on a page etc can be monitored in real time.
- The team can set up separate login pages for users and administrators since currently a toggle is used to differentiate between admins and users.
- The APIs for CastNXT, PlanNXT and EventNXT were not available so we created mock APIs for testing purposes.
- MockNXT was a mock app that we had created to test out OAuth and API integration. That can be removed in the future when all the NXT apps have been correctly integrated within CRMNXT.

Project Links

Heroku Deployment:

<https://fashionxt-crm-test.herokuapp.com/>

Pivotal Tracker:

<https://www.pivotaltracker.com/n/projects/2629149>

Source code:

<https://github.com/FashioNXT/FashioNXT-CRM-Service>

Project Demo and Poster Video:

<https://youtu.be/r9OgUaAusCo>