UNEST

CS 307 - Sprint #3 Retrospective

Team 13: Sudhanva Bharadwaj, Nivedha Kumar, Ram Laxminarayan, Grant McCord

What Went Well

Overall, the team completed all of the promised user stories for the third sprint. We finished the remaining sections of the user profile page. We also created a map of properties on the homepage that shows all of the properties that are not needed. We also created options to edit a listing, delete a listing, and change their password if they forgot it. We also made some changes to the UI to make it more consistent between pages. Finally, we got the messages page completely synced up with the database. These were all the main functionality of our page that wasn't yet completed. We finished everything that we had planned to do at the start of the semester and even more. Overall, our team collaborated very well together this sprint and we helped each other do the remaining tasks needed for our project.

Home Page:

1. As a potential tenant, I would like to see unique images for each of the properties on the homepage.

Task #	Description	Estimated Time	Owner
1	Add reference links to images of properties on the homepage. Figure out how to and display images using reference links from the database.	2 Hours	Nivedha
2	Create a unit test to make sure various images are shown on the homepage for different properties by pulling links to the images from the database.	1 Hour	Nivedha

Completed:

I added an array called photos to the listings collection in the database. I used the first image from this array to display in the property card. In the axios call for the property listing data, I get this additional piece of information to display. Previously this field was hardcoded, so every property had the exact same image. The reason why it's an array in the database is that the user can upload multiple images in the post new listing page. In the future, we would like to make this application show those multiple images in a carousel on the detailed property listing page.

2. As a potential tenant, I would like to click on a user profile icon on the homepage and see a drop down with options that route the user to other pages of the application.

Task #	Description	Estimated Time	Owner
1	Add an avatar icon to the top of the homepage	1 Hour	Nivedha
2	Add a drop down menu to the avatar icon so when the icon is clicked, a drop down options are displayed.	2 Hour	Nivedha
3	Add a routing feature so when the My Profile option is clicked, the user is routed to the user profile page.	0.5 Hour	Nivedha
4	Add a routing feature so when the Message option is clicked, the user is routed to the Messages page.	0.5 Hour	Nivedha
5	Add a routing feature so when the Post Listing option is clicked, the user is routed to the post a new listing page.	0.5 Hour	Nivedha
6	Add a routing feature so when the Logout option is clicked, the user is rerouted to the Logout page and the token in the cookie for the user is deleted.	2 Hour	Nivedha
7	Create a unit test to make sure that a user profile icon is displayed on the homepage and when clicked, the user can be rerouted to other pages of the application.	1 Hour	Nivedha

Completed:
I added an avatar icon picture to the header that was a picture of me. Then I added a dropdown menu in MUI. Unfortunately, the dropdown menu was blank probably due to using a combination of tailwind and MUI on the homepage. Therefore, I switched to using tailwind to create a drop down menu. I made all of the items links except for the logout option which instead was a button. However, the button did not look like the links at all, so I had to do a lot of styling to get it to look more like the links. I needed a button for the logout since in addition to linking to the login page, it also needs to delete the user’s token. Then, after I finished all the major functionality, I added user context so that the avatar icon showed a picture of the user that is currently logged in instead of the picture of me.

3. User Story 3: As a potential tenant, when I click on a property card, I would like to be taken to the corresponding property listing page with more detailed information for that property.

Task #	Description	Estimated Time	Owner
1	Implementing a routing feature so that when the property card is clicked it routes to the specific property listing. Send the property id that was clicked to that component	3 Hours	Ram
2	Fetch the specific property information for that property that was clicked	2 Hours	Ram
3	Display some information from the database such as name and address from values in the database.	3 Hours	Ram
4	Create a unit test to make sure that when a property card is clicked, the user is taken to the corresponding property listing page.	2 Hours	Ram

Completed:
On the homepage, when a user or potential tenant clicks on one of the properties on the home page, they would be redirected to a property listing page to the corresponding property that they clicked on with detailed information about the property. This was done using the useParams() method from react-router-dom, as the url for each property listing page was /propertylisting/id with id being the specific id of a specific property in the UNEST database in test.listings. All the information of each property listing was displayed by fetching the information for each property listing from the same database, which was done by using a .get method in index.js in the api folder, and then it was called in the propertyListings.jsx file with an axios call to ultimately fetch all the data and display it on the property listing page, which was done by storing all the data in a property variable.

User Profile Page:

1. As a potential tenant, I would like to see and edit my personal habits and roommate preferences on the user profile page.

Task #	Description	Estimated Time	Owner
1	Pull in data from the database for the personal habits section.	2.5 Hours	Nivedha
2	Pull in data from the database for the	2.5 Hours	Nivedha

	roommate preferences section.		
3	Using the pencil edit icon, implement functionality that allows users to edit and update the attributes in the personal habits section.	3 Hours	Nivedha
4	Using the pencil edit icon, implement functionality that allows users to edit and update the attributes in the roommate preferences section.	3 Hours	Nivedha
5	Create a unit tests to make sure the personal habits section and the roommate preferences section of the user profile page are all viewable and getting edited properly	3 Hours	Nivedha

Completed:

I made an axios call in the frontend that makes a http GET request to retrieve user information from the users collection in the database when the user profile page is mounted (started). It also retrieves that information anytime a change is made to the user's collection. It would then go to the corresponding app.get method in the backend index.js file by looking up the matching url. In that app.get method, I used the mongoose library User.findById() method to retrieve all the property information from the database. res.json then converts that response into a JSON object of key value pairs and sends it back up to the frontend. After retrieving that information I displayed it in two different sections on the user profile page: personal habits and roommate preferences sections.

2. As a potential tenant, when I click on the profile icon image, I would like to be able to upload a new profile icon image.

Task #	Description	Estimated Time	Owner
1	Pull profile icon image from the database to display on the screen	2 Hours	Nivedha
2	When the edit pencil icon is clicked, allow the profile icon to be clickable. Create a dialog that pops ups when the profile icon is clicked. Using that dialog, the user can upload a new image from their local machine.	2 Hours	Nivedha
3	Create a Multer storage system that uploads the image to the /uploads folder in the express/node server.	2 Hours	Nivedha

4	When the user clicks the save button, the profile image attribute is updated in the database and is shown on the screen.	3 Hours	Nivedha
5	When the user clicks the cancel button, the profile image returns to what it originally was.	2 hours	Nivedha
6	Create a unit test to make sure the avatar image icon appears and is editable correctly.	3 Hours	Nivedha

Completed:

I first created a profile_pic attribute in the users collection. Then I pulled that and displayed it in an avatar uploader component. Then, I wrote code for a dialog pop up that allows users to upload a new image from their local computer. Next, I wrote code for a multer storage system that temporarily stores the images in the express server. I wrote the code so that it pushes the image to the uploads folder when an image is chosen from the user's local machine. Then, when they hit the save button, I made an axios call that makes a http put request to update the profile_pic attribute in the database. If they hit the cancel button instead, I revert the image back to the original image. Finally, I implemented user context for this feature so that it shows the image of the logged in user.

Map:

1. User Story 9: As a potential tenant, I would like to be able to view a map for each property that I am interested in.

Task #	Description	Estimated Time	Owner
1	Ensure a map is centered around a specific property.	3 Hours	Ram
2	Include a marker and names, address for the specific property after clicking on the marker.	2 Hours	Ram
3	Include redirection to each map from a specific property listing.	2 Hours	Ram
4	Write a test to make sure the redirection redirects to the right map centered around the right property listing.	2 Hours	Ram

Completed: When a user clicked on the property image in a specific property listing page, they would be redirected to a map. This was done by passing the id of a specific property to redirect to a specific map to a specific property. This was performed by using useParams() in react-router-dom. On the map, a marker was displayed, which in this case was a pin which was incorporated by setting the icon attribute of the marker to a pin icon. The marker was displayed at a specific coordinate of the specific property based on its address, which was done using a fetch call to get the latitude and longitude coordinates of a specific property based on its address. Finally, when the marker was clicked, the name, address, and distance from a specific campus of the property that the map refers to is displayed, which is all fetched from the database using an axios call, and it is all stored in a property variable.

Messages Page:

1. User Story 10: As a user, I would like all existing chats to be displayed after I start a new chat.

Task #	Description	Estimated Time	Owner
1	Ensure that each message sent gets added to the database.	4 Hours	Ram
2	Ensure messages are fetched from the database based on who the sender/receiver is.	3 Hours	Ram
3	Link each chat to a specific chat such that clicking each chat redirects to a different chat with the right user.	2 Hours	Ram
4	When creating a new chat, that chat should then show up in the main messages page.	3 Hours	Ram
5	Write a test to ensure that a new chat is created and shows up/rendered in the main messages page.	2 Hours	Ram

Completed: When a user is logged in as a specific user and on the main messages page with url /messages, they can search a user who they want to message and send them a message. This is done by adding the messages to the database using an axios.post call, and then fetching the messages using an axios.get call and storing that data in a variable. This variable would be an array variable, so we would display the messages on the frontend by using variable.map and then displaying the specific contents of that variable. When a new chat was created, that chat would then show up in the page with url /messages, which was done by using and axios.get call to determine if there was a last message between two users, and then based on that, each chat would be displayed with its last message accordingly if there existed a last message between the two users, and each chat would link to the right user to message by determining the url using Link from react-router-dom. Finally, if I were to log out as the current user and log in as another user that we messaged as the previous user, we would be able to see the chat on the messages page and be able to click on that chat. The message would show up on the left side when we click on the specific chat when we initially click on the chat because received messages show up on the left, which was implemented through styling.

2. User Story 11: As a user, I would like to be able to delete a message that was sent.

Task #	Description	Estimated Time	Owner
1	Add frontend styling such that there is a delete option.	2 Hours	Ram
2	Ensure that when a user tries to delete a message, the message is no longer rendered on the screen.	3 Hours	Ram
3	Ensure that the message is also deleted from the database when the user deletes a message.	3 Hours	Ram
4	Write a test to make sure that the message the user deletes is no longer on the user’s screen or in the database.	2 Hours	Ram

Completed: When a user clicks on the last message that they sent, they would see a delete option pop up, which was done through styling and checking the index to ensure that it was the last message sent, and also by making sure that the current user is the one who sent the message. After that, when a user clicked on that delete option, the delete option would immediately remove from the frontend and from the database, which was done through using an axios.delete call to delete the message, and an axios.get call to refetch the messages.

Map Page:

1. As a potential tenant, I would like a map view to see where all of the properties are relative to campus.

Task #	Description	Estimated Time	Owner
1	Swap the listings and the map on the home page when a button is clicked.	30 minutes	Grant

2	Implement a map of Purdue University.	3 hours	Grant
3	Add markers to the map showing where all of the listings are.	8 hours	Grant
4	Add a marker to the map showing where the PMU is located.	30 minutes	Grant
5	Write test cases to make sure markers are properly updating and that the map works.	4 hours	Grant

Completed:

I implemented a map button on the home page that when clicked will swap whether the user can see the cards containing the listings or the map. The map defaults to a zoomed out map of Purdue University, centered on the Purdue Memorial Union. There are various markers on the map, by default there is one at the Purdue Memorial Union, otherwise there is a marker corresponding to each of the addresses of the listings in the database.

2. As a user, I would like the ability to click on a marker and view information about that property.

Task #	Description	Estimated Time	Owner
1	Make the markers clickable.	2 hours	Grant
2	Show a popup on marker click that will tell the user basic information about a property.	4 hours	Grant
3	Write test cases to make sure the markers are clickable, showing popups, and showing the correct data for each listing in the popup.	4 hours	Grant

Completed:

All of the markers that were implemented in the last user story are now clickable, and whenever a marker is clicked it will show a popup that will tell the user basic information about a listing. When the Purdue Memorial Union is clicked, the popup will simply say ‘Purdue Memorial Union’, but for any other marker when clicked it will show the name of the listing and its price, receiving all of this information from the database.

3. As a user, I would like the map to be user friendly with helpful features.

Task #	Description	Estimated Time	Owner
1	Make the map scrollable.	2 hours	Grant
2	Allow the user to zoom in or out of the	3 hours	Grant

	map.		
3	Only show the markers that the user has searched for.	4 hours	Grant
4	Write test cases to make sure the map is scrollable and that only the searched markers appear on the map.	4 hours	Grant

<u>Completed:</u>

The user can use the scroll wheel to zoom in or out of the map, and can click and drag the map to scroll. When the user inputs values into the search bar, the searched terms will be reflected in the markers that are shown on the map. The same terms available for listings are available for markers, so the user can search for a university name or a property name.

Forgot Password Page:

1. As a user, I would like to see my password when I forget it.

Task #	Description	Estimated Time	Owner
1	Route the “Forgot Password” button on the sign in page to a new form.	30 Min	Sudhanva
2	Create a form that asks the user for their email and their birthday, as well as a question to look for their account and verify it is them.	2 Hours	Sudhanva
3	When the form is submitted, the answers should be verified and if they match, the user’s password should be displayed.	3 Hours	Sudhanva
4	Write a test to ensure that clicking on the forgot password button routes and render’s a new form.	1 Hour	Sudhanva
5	Write a test case to verify that the password associated with the email is only shown with a valid response to the “secret” question”.	1.5 Hours	Sudhanva

<u>Completed:</u>

The user is able to change their password once they properly enter their email and username. Once those 2 fields are validated, a secret answer is required to further validate the user. If all 3 fields match an existing user in the database, a new form is loaded prompting the user to enter a new password. The user can then submit the form which will change the password for that user in the database. Finally, the user is routed to the login page where they can use their new login.

Property Listing Page:

1. As a user, I would like to edit a listing that I have previously posted.

Task #	Description	Estimated Time	Owner
1	Create an edit button next to each of the properties listed by the user.	2 hours	Sudhanva
2	Route the edit button to the property listing form. This time, the fields of the form should already be filled out with the existing info.	2.5 Hours	Sudhanva
3	Add all the fields from the “new listing” form and also add more info such as number of bedrooms and bathrooms. Create the form with the same styling as the original.	3 Hours	Sudhanva
4	Write a test case to verify that the fields on the edit form match the ones of the original listing in the database.	1.5 Hours	Sudhanva
5	Write a test case to verify when the edit form is submitted, the database is modified with the edited fields.	1 Hour	Sudhanva

Completed:

The feature was fully implemented and tested to ensure that a user can edit all fields of their original listing. If logged in for the account the post was listed, a user is able to click on it and have options to edit or delete the listing. Once the edit is chosen, a new version of the original property form is loaded with the fields autofilled from the database for the listing they clicked on. The user is able to make changes to one or many of the fields and hit save. Once saved, the data is sent to the database and updated on all the pages where the data is imported.

2. As a user, I would like to add more details that are specific to my property, such as the number of bedrooms and bathrooms.

Task #	Description	Estimated Time	Owner
1	Create fields for number of bedrooms and bathrooms in the new property form.	1.5 hours	Sudhanva

2	Add the entries from these fields (Number) to the database and modify the Place Schema to reflect the changes.	2.5 Hours	Sudhanva
3	Add a field to enter the closest college/university campus to the property.	1.5 Hour	Sudhanva
4	Add in a field for the distance from campus and add in the field into the database.	1 Hour	Sudhanva
6	Write a test case to verify that the newly added fields properly post to the database when the form is submitted.	1.5 Hours	Sudhanva
7	Write a test case to ensure that the added fields can also be edited via the edit form.	1.5 Hours	Sudhanva

Completed:

The user is able to add more detail into their listing that was outlined in the story. All the fields that were promised were added to the listing form. The data was organized into the collections of the database where info can be pulled. The miles from campus was used to be displayed on the main card of the homepage.

3. As a user, I would like to delete an old listing.

Task #	Description	Estimated Time	Owner
1	Create a button that says “Delete” next to each property on the “My Properties” page.	2 Hours	Sudhanva
2	Setup a remove method that finds the property that was selected to be deleted and removes it from the collections of the database.	2.5 Hours	Sudhanva
3	Create a alert message on the “My Properties” page that displays the response from the remove method.	1.5 Hour	Sudhanva
4	Create a new page where user can choose between deleting and editing a listings. Page needs to be unique per post	2 hours	Sudhanva
5	Write a test case to verify that the chosen property is removed from the	1.5 Hours	Sudhanva

	database.		
6	Write a test case to ensure that the delete button is clickable on the “My Properties” page.	1 Hour	Sudhanva

Completed:

The story was completed with all the tasks completed as promised. The user is able to click on any of their past listings which takes them to a modify page. On that page, there is a button to delete the listing. When clicked, the listing that was originally clicked is deleted from the database and no longer shows up in the user’s previous listings. This listing can no longer be found on the mainpage as well which is expected when deleting the listing.

What did not go well?

For this sprint, the team completed all the promised user stories. Overall, the sprint can be declared as successful but there is still room for improvement. The team would like to improve on small bugs and issues that impact the overall consistency and usability of the site. While all the planned features have been implemented, it would be beneficial to go back and fix the styling for the different pages. Since each member on the team developed pages on their own, locations of different buttons vary from page to page. One example of this is the “home” button which is located on every page to route the user to the mainpage. On some pages, the button is located on the top left while others have it on the top right. It would help the consistency and overall user experience of the site to place the button on the same spot across all the pages.

Developer Story 1 (numbering from the revised sprint planning doc): As a potential tenant, I would like to see and edit my personal habits and roommate preferences on the user profile page.

Task #	Description	Estimated Time	Owner
3	Using the pencil edit icon, implement functionality that allows users to edit and update the attributes in the personal habits section.	3 Hours	Nivedha
4	Using the pencil edit icon, implement functionality that allows users to edit and update the attributes in the roommate preferences section.	3 Hours	Nivedha

Completed, but can be improved:

I did all the requirements I specified in the user story above. One change I would have liked to make is to add suggestions for what the user should type in each field. For instance, it might not be clear to the user what they can write for the “Sleeping Type” field.

Developer Story 10: As a user, I would like all existing chats to be displayed after I start a new chat.

Task #	Description	Estimated Time	Owner
1	Ensure that each message sent gets added to the database.	4 Hours	Ram

2	Ensure messages are fetched from the database based on who the sender/receiver is.	3 Hours	Ram
3	Link each chat to a specific chat such that clicking each chat redirects to a different chat with the right user.	2 Hours	Ram
4	When creating a new chat, that chat should then show up in the main messages page.	3 Hours	Ram
5	Write a test to ensure that a new chat is created and shows up/rendered in the main messages page.	2 Hours	Ram

Completed, but can be improved: All of the requirements for this user story are done, as all of the tasks listed here are complete and the messages page has full functionality. However, there are times where the messages do not always show up properly until the page is reloaded when viewing a specific chat, as the messages will always be shown, but sometimes not until the page is reloaded. Therefore, this user story has all of its requirements completed, but it could be improved with making the messages render faster.

1. As a user, I would like the map to be user friendly with helpful features.

3	Only show the markers that the user has searched for.	4 hours	Grant

Completed, but can be improved:
All of the requirements for this user story have been implemented, but I was unable to implement continuous updating in the map search the same way I implemented it in the property listing search. Instead, whenever the updates to the filters are only shown the first time the user opens the map, changing the filters while in the map doesn’t modify the markers shown at that time, and requires the user to reopen the map to see the filter changes.

How should we improve? (future improvements)

Upon completion of the third and final sprint, the team feels that we met all the expectations and promises made at the start of the project. We completed key components of the site and ensured that their functionality closely aligns with the final expected design. The team did a great job of communicating throughout the sprint and seeking help when needed. Almost all the tasks from the backlog were picked up and completed in terms of functionality which the team feels really proud of. The overall UI of the site looks exactly as planned and the core functions and features are implemented which shows the capabilities of the team. While this course and the project is coming to an end, the team plans to further develop the site as a personal project and ideally get it to a point where it can be released for others to use.

The team really believes that this site fills a gap for students that are looking for temporary housing across the nation. With a great foundation as the team has built, the site can be fully developed and used by others which would be the ultimate goal for this application. Before this application can be released, there are some bugs and issues that must be fixed. One of the biggest features that was not implemented was the booking functionality. Currently, the site does not handle payments and secure rooms for user's which is an important feature. Third party APIs can be used to handle these payments such as square and stripe. The team plans to research more into the implementation of these tools so that the application is closer to deployment.

Another major task that still needs to be done by the team is thorough testing of the application. The team already has a list of bugs that need to be addressed but is sure that there are many more in the applications that need to be worked out. Ideally, the implementation of continuous delivery would aid the team and allow for more efficient testing. One bug the team has identified is with the upload photos feature where a user can link to their property listing. The feature allows for a user to upload one or many photos for their listings which is then displayed on the main page. However the team realized that deleting images of listings is quite important and would need to be implemented for a better user experience with the application.

Overall, the team is satisfied and proud with the current state of the project. The designed application and features have been implemented to demonstrate the concept of the site and its necessity. Additionally, the team has shown their collaboration and technical skills with this project which can be applied in the industry.