UNEST

CS 307 - Sprint #3 Planning

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

Sprint Overview:

For this sprint, the team plans to complete the core functionality of the website. The user login and browsing process will replicate other booking sites with the ability to browse listings and look into specific ones. By the end of the sprint, the website needs to have all the pages developed and functional. All buttons and interactive features need to route to other pages or display some kind of change to the user. The ability to message a user and refer back to old chats will also be complete by the end of the sprint which will require the chats to be stored in the database. Additionally, the ability to view the properties near a campus on a map will be added as it can provide a clear oversight of all the properties within a user's area. Finally, the team will work to establish a consistent front end design amongst all the pages which will allow for the website to look like a completed product.

Scrum Master:

The scrum master/team leader for our group is Nivedha Kumar.

Scrum Meeting Schedule:

The regular meeting times will be Tuesday 8:30 pm - 10:00 pm, Thursday 8:30 pm - 10:00 pm, and Saturday 5 pm - 6:30 pm.

Risks and Challenges:

We have addressed all major challenges during the last sprint so it is essential that we continue to reduce the chances of risks and communicate before making any major changes. With the final deadline coming up for the project, we want to do our best in maintaining clean and working code on our repository.

We also will be focusing on completing the integration of the database with our application. While the team is familiar with the process, there are a lot of data points and types which can cause errors if not organized correctly. In the previous sprint, we initially had some issues with displaying some of the stored data to the user. In particular, images that were uploaded for a listing were no re rendering on other pages. That issue has since been fixed but the images still need to be properly formatted so that they design and layout of the page is not altered. Currently, the text is pushed out of the way to render the images and the images that are shown are all different in size. Another foreseeable challenge for this sprint will be the implementation of the map. The map is one of our essential features of the site and developing this will require the team to learn new skills and techniques.

Current Sprint Detail:

Total Story Hours:

User	Total Hours
Sudhanva Bharadwaj	38 hours
Ram Laxminarayan	43 Hours
Nivedha Kumar	42.5 Hours
Grant McCord	37 hours

User profile:

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 roommate preferences section.	2.5 Hours	Nivedha
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

Acceptance Criteria:

- Given that the user information is pulled from the database correctly, the personal habits section should be shown on the user profile page
- Given that the user information is pulled from the database correctly, the roommate preferences section should be shown on the user profile page
- Given that the edit option is functional for the personal habits section, the users can edit and update attributes for their personal habits
- Given that the edit option is functional for the roommate preferences section, the users can edit and update attributes for their roommate preference
- Given that the user information is pulled from the database correctly, when the user goes to the user profile screen, the unit test cases should run successfully.
- Given that the user information can be edited properly, when the user goes to the user profile screen, the unit test cases should run successfully.

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

Acceptance Criteria:

- Given that the avatar icon is functional, when the edit pencil icon is clicked, the avatar icon should be clickable
- Given that the avatar icon is functional, when the avatar icon is clicked, it should pop up with a dialog that allows the user to select a new image from their local machine.
- Given that the avatar icon is functional, when the save button is clicked, the profile image attribute should update with the new image in the database and show that new image on the screen.
- Given that the avatar icon is functional, when the cancel button is clicked, it should return to the image that it previously was.
- Given that the avatar icon is functional, the unit test cases that test if the avatar icon properly uploads new images should be successful.

Homepage:

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

Acceptance Criteria:

- Given that the routing is implemented correctly, when a property card on the homepage is clicked, the application should route the user to the corresponding property listing information page
- Given the property information is pulled from the database correctly, when a property card is clicked, property listing page should fetch fields from the database
- Given the property information is pulled from the database correctly, when a property card is clicked, the property listing page should show some fields from the database such as name and address on the page.
- Given that routing and pulling information from the database are functional, when the property card is clicked, the unit test case should be successful

4. 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

Acceptance Criteria:

- Given that the database for the homepage is updated, there should be an photos attribute in the users collection in the mongodb database
- Given that the application properly fetches the property's image from the database, the image should be displayed in the property card on the homepage
- Given that properties are shown on the homepage, the unit test case displaying different images for the homepage should be successful

5. 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	1 Hour	Nivedha

	user can be rerouted to other pages of the application.		

Acceptance criteria:

- Given that the avatar icon is added, when the user views the homepage, then they can see their avatar icon at the top right of the page.
- Given that the avatar icon is clickable, when a user clicks on their avatar, then they will see the drop down options.
- Given that the dropdown options are clickable, when a user clicks on the my profile option, then the user can see their own profile.
- Given that the dropdown options are clickable, when a user clicks on the view messages option, then the user can see the messages page.
- Given that the dropdown options are clickable, when a user clicks on the post listing option, then the user can see the post a new listing page.
- Given that the dropdown options are clickable, when a user clicks on the logout profile option, then the user gets sent to the login page and the token in the cookie is removed.
- Given that the dropdown options are all different, when a user clicks on one option and then the other, then the user will be routed to a different page based on the specific dropdown option.
- Given that there is a avatar icon on the homepage, when the user clicks on it and then clicks on one of the options in the drop down menu, the unit test case should be successful

Map:

6. 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

Acceptance Criteria:

- Given that the new page is correctly implemented, when the user clicks the map button then the map page should appear.
- Given that the new page is correctly implemented, when the new page appears then a map should be loaded onto the screen.
- Given that the map is correctly implemented, when the user is on the map page then the map should show Purdue University.
- Given that the map is correctly implemented, when the user looks at the map it should show the location of the PMU.
- Given that the map is correctly implemented, when the user looks at the map it should show the locations of the listings.

7. 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

Acceptance Criteria:

- Given that the markers are correctly implemented, when the user hovers over the marker then it should be clickable.
- Given that the markers are correctly implemented, when the marker is clicked then a popup should appear.
- Given that the markers are correctly implemented, when the popup appears it should be populated with correct listing information.

8. 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 map.	3 hours	Grant
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

Acceptance Criteria:

- Given that the map is implemented properly, when some input is obtained then the user should be able to scroll the map.
- Given that the map is implemented properly, when some input is obtained then the map should zoom in.
- Given that the map is implemented properly, when some input is obtained then the map should zoom out.
- Given that the markers are implemented properly, when there is some criteria in the search box then only the markers that match that criteria should be shown.

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

Acceptance Criteria:

- Given that the map is displayed, when a user views the map, then they can see the specific property they were interested in centered on the map.
- Given that a specific map is centered around a specific property listing, when a user views the map, then they will see a marker at the location of the specific property.
- Given that there is a marker displayed on the map, when a user clicks on the marker, then they will be able to see attributes of the property such as the name and address.
- Given that redirection is implemented, when a user views a specific property listing, then they can redirect to a specific map from that property listing.

- Given that there exists multiple property listings, when a user views one property listing and then another, then they can see distinct maps representing the different property listings.

Messages

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

Acceptance Criteria:

- Given that each message a user sends is added to the database, when the database is viewed, then the message will be shown in the database along with other attributes such as the sender and receiver.
- Given that messages are fetched from the database, when the user views their screen, then they should be able to see all their messages displayed on their screen appropriately based on if they are the sender/receiver.
- Given that each chat is unique to a specific user, when a user clicks/views a chat, then they are linked to the right chat and view the person name/username that they wanted to chat with.
- Given that a user chats with a new user, when they go back to the main messages page, then they can view that new chat existing on that page.
- Given that the user is redirected back to the main messages page after messaging a new user, when they view the main messages page, then all of the other chats shown previously will still be displayed.

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

Acceptance Criteria:

- Given that there is a delete option, when a user clicks on the last message they sent, then they will see deleting the message as an option.
- Given that a delete option exists, when the last sent message is clicked by a user, then the delete option will be shown below the message the user clicked on.
- Given that there is a delete option, when a user clicks on the delete button, then the user would no longer see the message they deleted on their screen.
- Given that the user deleted a message, when the database is checked, then the message is no longer shown in the database.

Forgot Password:

12. 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

Acceptance Criteria:

- Given that there is a “Forgot Password” button, when a user clicks it, they should be routed to a form that asks for their email and username as well as a secret word.
- Given that the forgot password form renders, when a user clicks on a field, they should be allowed to type in their responses.
- Given that a user has properly filled out the form with valid responses, when the submit button is clicked, an alert should appear reminding the user of their password.
- Given that a user fills out the form with invalid responses, an alert should appear asking the user to try again.

Property Listing Page:

13. 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

Acceptance Criteria:

- Given that the user has created a listing, when the user opens the “My Properties” page, a edit icon should appear next to each of the entries.
- Given that the edit button exists, when the user clicks it, the edit property form should appear with the current fields of the property filled out.
- Given that the edit listing form exists, when a user opens the form, a get request should be made to the database to retrieve the exists fields related to the property.
- Given that the edit listing form exists, when the user submits the form, a post request should be made to the database to edit the modified fields.
- Given that the edit listing form exists, when the user scrolls through the form, an option to enter the number of bedrooms and bathrooms should appear.

14. As a user, I would like to add more details that are specific to my property, such as 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

Acceptance Criteria:

- Given that the new property listing form properly renders, when a user scrolls through the form, fields for the number of bedroom and bathrooms should appear and require a number entry.
- Given that the new property listing form works correctly, when the user submits the form, they should be redirected to the main page and a success message should be shown.
- Given that the listing page is created as designed, when the page is loaded, all the fields should be viewable with a text box or select box below each of the fields.
- Given that a listing question needs a user response, when a user clicks on the text box, they should be able to enter with their keyboard and the response should be saved until the form is submitted.
- Given that a user has typed into a text box, when the entry is invalid, an error message should be displayed asking the user to try again.
- Given that the database is connected to the registration form, when the user clicks submit the data should be transferred into the database with a new entry and id for the listing created.

15. 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 database.	1.5 Hours	Sudhanva
6	Write a test case to ensure that the delete button is clickable on the “My Properties” page.	1 Hour	Sudhanva

Acceptance Criteria:

- Given that the “My Properties” page is working correctly, when the user clicks a listing, a delete button should appear for each property listing.
- Given that the delete button exists, when the user clicks the button, a remove method should be executed and the entry should be removed from the database.
- Given that the delete button exists, when the user clicks the button, a alert message should appear when the listing is successfully removed.
- Given that the database is working properly, when the user delete a listing, the property should not appear in the list of properties on the homepage.

Backlog:

User Registration

~~As a user, I would like to be able to make an account by filling out a user registration form.~~

As a user, I would like to be able to login with my username & password.

~~As a user, I would like my password to be hidden when I type it.~~

~~As a user, I would like to verify my password before creating my account.~~

As a user, I would like to be able to change the university associated with my account to a different university.

~~As a user, I would like to be able to view and update my user bio.~~

~~As a user, I would like to change and reset my password.~~

As a potential tenant, I want the site to verify that listers are students by confirming their school email.

As a lister, I want the site to verify that any potential tenants are students by confirming their school email.

~~Home Page~~

~~As a user, I would like to see a grid view of multiple properties in my selected area.~~

~~As a user, I would like a map to show all properties available in my selected radius.~~

~~As a user, I would view the name of the property, price, dates available for rent, and miles from campus of each property.~~

~~As a potential tenant, I would like to see all the available property listings with images and their location, date range, and cost.~~

~~As a potential tenant, I would like to filter the properties based on the room of a house or an apartment.~~

~~As a potential tenant, I would like to filter the properties by specifying a given price range.~~

~~As a potential tenant, I would like to filter the properties by a certain number of beds and bathrooms.~~

~~As a lister, I would like to have a button to create a new post.~~

~~As a lister, I would like to access my inbox with a single button.~~

~~As a potential tenant, I would like to have a button to view all of my favorite properties.~~

~~As a user, I would like the ability to search for properties on different campuses.~~

~~As a user, I would like to be able to see various types of properties located in my area (added this user story in this sprint)~~

~~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 that is pulled from the database.~~

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

~~As a potential tenant, I would like to click on a user profile icon on the homepage and see a drop down with the options to go to the user profile page or logout page from there.~~

~~Listing information page (made for potential tenants only)~~

~~As a potential tenant, I would like to be able to view specific information about a property as described in features 26 and 27 below.~~

~~As a potential tenant, I would like to have the option to favorite properties so I can view them later.~~

~~As a potential tenant, I would like to be able to see the locations of the properties.~~

~~As a potential tenant, I would like to see the listers' contact information.~~

~~As a potential tenant, I would like to directly start a chat with the lister.~~

As a potential tenant, I would like to see that the lister is a verified student.

Lister's Own Listings Page (made for listers only)

~~As a lister, in the form, I would also be able to provide the property's exact location (will be shown in a map to the potential tenant) and pictures of the property to be displayed at the top of the page.~~

~~As a lister, I would like to be able to submit the information in the form and see it posted in the homepage listings.~~

~~As a lister, I would like to be able to view my posted listings on the page.~~

~~As a lister, I would like to be able to update my posted listings.~~

As a lister, I would like to see that the potential tenant is a verified student.

Message System

~~As a user, I would like to easily view all my messages.~~

~~As a potential tenant, I would like to be able to contact listers/users about their property.~~

~~As a user, I would like all existing chats to be displayed after I start a new chat.~~

~~As a user, I would like to be able to delete a message that was sent.~~

~~As a lister, I would like to have my messages to other listers and my messages from potential tenants be separated.~~

~~As a potential tenant, I would like listers to have the names of their properties included to help differentiate them.~~

Map

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

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

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

~~User Page:~~

~~As a user, I would like to see a user profile page with information about myself.~~

~~As a user, I would like to be able to edit information in the user profile page.~~

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

Non Functional

Architecture and Performance
We will be using the MERN framework: Mongo.db, Express.js, React, and Node.js. We will develop frontend components using React. React is a library that reacts to changes in the data and re-renders the UI accordingly. The middle of the tech stack will be Express.js and Node.js. Express.js is a server-side web framework, while Node.js is a JavaScript server platform. We will be using Mongodb Atlas to host the database in the cloud, to avoid the maintenance issues with hosting the database on our local machines.

To make our website as efficient as possible, scheduled tasks will be incorporated to fetch data from the database and update the front end. The plan right now is to incorporate a scheduled task for every 2 minutes so that listings are automatically updated on the home page. Since a chat feature is a critical part of the application, efficient use of calls will be focused on to reduce load times.

List of Performance Features:

1. The page should load all the content within 5 seconds of a refresh
2. The site should use a scheduled task to update the listings every 2 minutes
3. A listing can be posted or updated 24 hours a day.
4. Users can contact customer service for help with technical difficulties 24 hours a day.
5. Chats should refresh at most 20 seconds after a message is sent.

Security
Security will be a major point of emphasis as the site will be dealing with people's personal information. We will be hiding personal information behind user authentication to help protect privacy. Since we are focusing on students, all users will need to be affiliated with a university. Accounts will be verified through an authentication API and only verified users will have access to certain features such as messaging and viewing contact info.

We will be using an authentication API, such as Auth0. For instance, Auth0 provides a React SDK, which would allow us to implement Auth0 authentication and authorization in React apps.

1. We plan to use Auth0, an authentication API, to authenticate student emails.
2. Auth0 will connect to our database to securely store identity information.
3. Auth0 will give us extensibility and security capabilities such as brute force and suspicious IP throttling.
4. Auth0 will give us up to 7,500 monthly active users and unlimited logins.

Usability
Our goal is to make the website easy to use while still giving the user all the information about the property that they need. We want to include a main home page for all users that will show all available apartments to lease, with the ability to filter by various features like location, price, etc. We also need an additional option for current tenants to easily input information about their property. The website will also feature an admin view for property managers to view contracts

and manage renters. Finally, we want to ensure our application is viewable and adjusts to various screen sizes from a laptop screen to a large screen monitor.

Hosting/Deployment

We plan to just use GitHub and develop locally as we work through the project, and then we will deploy it and make one final release after we have the main functionality of the site done. We plan to host our MongoDB server on the cloud.