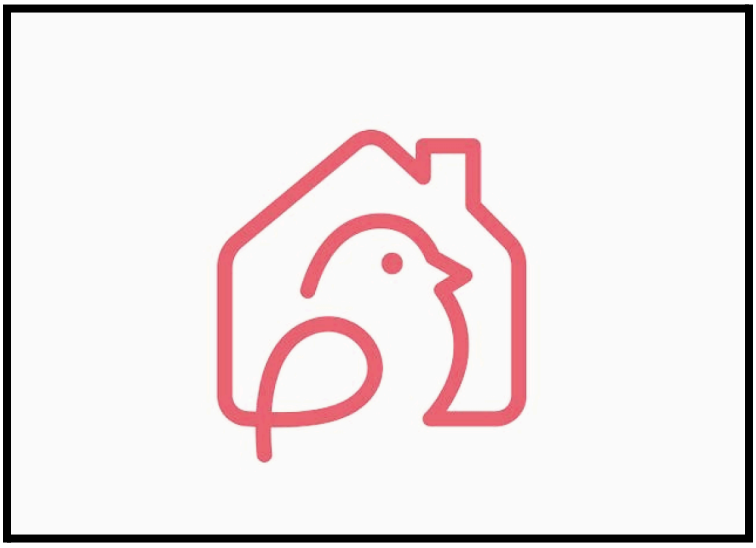
UNEST

CS 307 - Sprint #1 Planning

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



Sprint Overview:

For this sprint, the team aims to layout the frontend for the main pages of the website. The team will use the designed front end pages as a guide to create the components in React. We will aim to set up the database which is the backbone of our website. Required fields will be identified and the attributes will be created. With the completion of this sprint, we hope to have most of the front end layed out, including the home page, which will help with the creation of routes later in the project. Finally, we hope to complete most of the registration process so that user accounts can be created with appropriate parameters which are used to populate other pages.

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:

Since this is the first sprint, we believe that setting up the environment and getting familiar with the languages will be the main challenges. Most of the members in our group are not familiar with React, Express.js, and Node.js, so it will take a longer time to get accustomed to using these languages. To resolve this, the team will meet to practice the skills via tutorials and online exercises.

Also, creating a database schema that efficiently stores all the users data could be challenging. We want to strike a balance between not storing repetitive data but also ensure the SELECT queries are not too complicated. SELECT queries are not complicated when no JOINS are needed and all the information is already combined and stored in one table. However, this leads to repeating data and therefore, sometimes it is better to break that table into multiple tables to avoid repetition and have better organization.

Current Sprint Detail:

Authentication Page:

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

Task #	Description	Estimated Time	Owner
1	Create button components for login and sign up	2 hours	Sudhanva
2	Create a front end form with all required fields including name, email, and school.	2.5 hours	Sudhanva
3	Make Submit button appear once all fields have been validated	2 hours	Sudhanva
4	Create user entry boxes and drop down menus for questions.	3 hours	Sudhanva
5	Connect responses from form to field of the database.	6 hours	Sudhanva
6	Test the functionality of the registration page and ensure all fields are available.	2 hours	Sudhanva

Acceptance Criteria:

- Given the Button components are created as expected, when clicked they should print a simple message on screen to display a successful click.
- Given that the registration page is created as designed, when the page is loaded, all the fields should be viewable with a text box below each for field.
- Given that a registration 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 submit button is created, when the form is complete, the “Submit” button should appear and be clickable.
- 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 user.

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

Task #	Description	Estimated Time	Owner
1	Create a text box for password entry	1 hour	Sudhanva
2	Replace the text of user entry with “*” character using password type and test.	1 hour	Sudhanva
3	Create a related field in database and ensure user entry is properly saved	2 hours	Sudhanva
4	Test the text box to make sure that the characters are represented as “*”	2 hours	Sudhanva

Acceptance Criteria:

- Given the text box is functional, when the user clicks on the text box then they should be able to type into it.
- Given the text box is functional, when the password is being typed then the characters in the box are masked with “*”.
- Given the text box is functional, when the user has entered a password and submits form then the password field in the database is properly updated to the corresponding user.
- Given that the user has created a password, when the form is submitted, the input password should be stored in the database.

3. As a user, I would like to verify my school email before signing up.

Task #	Description	Estimated Time	Owner
1	Create text-box to enter school email	2 hour	Sudhanva
2	String compare the entered email for “.edu” ending	2 hours	Sudhanva
3	Display a confirmed message or icon for valid email	1 hour	Sudhanva
4	Test registration with valid and invalid email and ensure that a warning message is displayed for invalid entries.	1.5 hours	Sudhanva

Acceptance Criteria:

- Given that the school field is properly created, when the user enters a valid “.edu” email, a success message or icon will appear.
- Given that the email authentication is properly configured, when the user enters an invalid email address, then an error message will be displayed and the user will be asked to enter a proper email address.
- Given that the user enters a valid email address, when the user finishes entering the login form, then they will see a confirmation message.

Database:

4. As a user, I would like my information to be organized and stored in the database

Task #	Description	Estimated Time	Owner
1	Set up MongoDB Atlas and create a database in MongoDB Atlas	4 hours	Nivedha
2	Write out a list of tables and draw a diagram to figure out their relationship to each other	6 hours	Nivedha
3	Write out the attributes for each table and their data types	4 hours	Nivedha
4	Determine the foreign keys for each table	2 hours	Nivedha
5	Create the design schema for each table on MongoDB	4 hours	Nivedha
6	Use Postman to test if queries to the database work and if the database is correctly storing data	4 hours	Nivedha

Acceptance Criteria:

- Given that the database is implemented properly, when I click on the database in MongoDB Atlas, then the database should be set up and viewable.
- Given that the database is implemented properly, when I click on the database in MongoDB Atlas, then the database should have all the tables needed to store information needed for our application.
- Given that the database is implemented properly, when I click on each table, then I should be able to see all the fields listed with their corresponding data types
- Given that the database is implemented properly, when the developers on our team edit the database schema for the tables, then those changes should be reflected in the database.
- Given that the database is implemented properly, when the user queries from the database, then they should get a correct response back.

Home Page:

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

Task #	Description	Estimated Time	Owner
1	Create a default square placeholder which will hold the information for all property listings with space for the image and the text.	3 hours	Nivedha
2	Find the right grid size for all the property listings and make the grid	3 hours	Nivedha
3	Create a unit test to verify that the grid is properly displayed	1 hour	Nivedha

Acceptance Criteria:

- Given that the home page is viewable, when a user looks at the homepage, then they should see a square/card placeholder for a property listing.
- Given that the homepage of the UI is fully formatted, when the user views the UI, then they should see a properly formatted division for the image and for the text below
- Given that properties are posted on the home page, when the user views the home page, then they should see the properties formatted in a grid with enough space between each other.

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

Task #	Description	Estimated Time	Owner
1	Write the name of the property in the info below the image and format it so it fits for all cards	0.5 hour	Nivedha
2	Write the dates available for rent below the property name	1 hour	Nivedha
3	Display the price for rent	0.5 hour	Nivedha
4	Write the miles from campus on the right side of the box	1 hour	Nivedha
5	Write a unit test to make sure that property, price, dates available for rent, and miles from campus of each property are all displayed	1 hour	Nivedha

Acceptance Criteria:

- Given that a property is posted on the homepage, when the user views the homepage, they will see the property name listed below the image.
- Given that a property is posted on the homepage, when the user views the homepage, they will see the dates available of the property listed below the property name.
- Given that a property is posted on the homepage, when the user views the homepage, , they will see the price for rent displayed for each property.
- Given that the property is posted on the homepage, when the user views the homepage, they will see a property's miles from campus for each property.
- Given that multiple properties are on the homepage, when the user visits the homepage, they will see the corresponding property name, price, date, and distance from campus for each property.

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

Task #	Description	Estimated Time	Owner
1	Organize how the information will fit in each square.	3 hours	Grant
2	Set up a way to get all of the property information from the database.	4 hours	Grant
3	Populate each square with all given information for the property.	5 hours	Grant
4	Add the ability for each square to be clicked	1 hour	Grant
5	Create unit tests to verify that the squares are correctly populating with information.	3 hours	Grant

Acceptance Criteria:

- Given the organization is implemented properly, when the squares are viewed then all of the information should be easily visible in each square.
- Given the queries are implemented properly, when the home page is viewed the web app should receive all of the property information from the database.
- Given the squares are implemented properly, when the home page is brought up then all of the squares should be populated with information.
- Given the squares are implemented properly, when a square is clicked then it should give off some signal that it registered the click.
- Given the squares are implemented properly, when the website is scrolled the squares should be repopulated as needed.

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

Task #	Description	Estimated Time	Owner
1	Add a button that says “Create Post”.	1 hour	Grant
2	Only show the button when the user is logged in.	1 hour	Grant
3	Have the button as a part of a header that doesn’t scroll so it’s always clickable.	1 hour	Grant
4	Test the button to make sure it works visually and on click.	1 hour	Grant

Acceptance Criteria:

- Given the button is implemented correctly, when I view the home page and am logged in then I can see the button.
- Given the header is implemented correctly, when I scroll down the page then the button is still clickable.
- Given the button is implemented correctly, when I look at the button then it should say “Create Post”.
- Given the button is implemented properly, when I click the button then it should show that the button was clicked.

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

Task #	Description	Estimated Time	Owner
1	Add a button that looks like an envelope.	1 hour	Grant
2	Only show the button when the user is logged in.	1 hour	Grant
3	Have the button as a part of a header that doesn’t scroll so it’s always clickable.	1 hour	Grant
4	Test the button to make sure it works visually and on click.	1 hour	Grant

Acceptance Criteria:

- Given the button is implemented correctly, when I view the home page and am logged in then I can see the button.
- Given the header is implemented correctly, when I scroll down the page then the button is still clickable.
- Given the button is implemented correctly, when I look at the button then it should look like an envelope.

- Given the button is implemented properly, when I click the button then it should show that the button was clicked.

10. As a potential tenant, I would like to be able to search for a property.

Task #	Description	Estimated Time	Owner
1	Create a search bar in the header of the web app.	1 hour	Grant
2	Allow the user to type information into the search bar when it is clicked.	1 hour	Grant
3	Write an SQL query to get all properties that match the property name that was searched.	4 hours	Grant
4	Repopulate each square with all properties that match the searched for property name.	2 hours	Grant
5	Create unit tests to verify that the SQL query works.	3 hours	Grant

Acceptance Criteria:

- Given the search bar is correctly implemented, when the search bar is clicked then a user should be able to add information.
- Given the search bar is correctly implemented, when the user enters information then it should query the database searching for properties that match that information.
- Given the query is correctly implemented, when the query is run it should return the matching information from the database.
- Given the home page is correctly implemented, when the home page receives the information from the query it should repopulate the squares with the given properties.
- Given the squares are correctly implemented, when the squares are repopulated then they should be clickable.

Listing Page:

11. As a potential tenant, I would like to be able to view specific information about a property as described in features 40 and 41 below.
    a. As a lister, I would like to be able to fill out a form with all the specific information about a property including paragraph description of the property and a list of amenities.
    b. 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.

Task #	Description	Estimated Time	Owner
1	Write out all attributes needed to be displayed for a property.	1 Hour	Ram
2	Create squares to organize where each information for each property will be displayed.	1 Hour	Ram
3	Import sample data of property information into frontend of the UI.	1 Hour	Ram
4	Create a square where the property image will be displayed.	1 Hour	Ram
5	Import sample property image.	1 Hour	Ram
6	Add a button in the frontend of the UI that has the purpose of allowing potential tenants to send a message to the property owner.	2 Hours	Ram
7	Add a button in the frontend of the UI that has the purpose of allowing potential tenants to schedule a tour of the apartment.	2 Hours	Ram
8	Create a test to ensure that buttons are clickable and redirect to the right page.	3 Hours	Ram
9	Display potential roommates from the property and message button under each roommate at the bottom of the page.	4 Hours	Ram

Acceptance Criteria:

- Given that the UI is formatted correctly, when the user views the listing page, they should be able to see details of each property attribute.
- Given that a button is displayed in the UI, when the user clicks on the button, they should be redirected.
- Given that the image of the property is displayed, when the user clicks on the image, they should be able to view the property image more closely.
- Given that the roommates are displayed at the bottom of the page, when the user clicks on the message button beneath the roommates name, they should be redirected to a new page.
- Given that the roommates names are displayed, when the user clicks on their name, they will be redirected to a new page.

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

Task #	Description	Estimated Time	Owner
1	Determine color to use to indicate that property is favorited.	1 Hour	Ram
2	Determine location of like button	1 Hour	Ram
3	Implement how the like button is toggled when liking and unliking a property.	4 Hours	Ram
4	Ensure that property is still favorited after refreshing.	2 Hours	Ram
5	Create a test to ensure that the like button is clicked and stays clicked (does not disappear after refreshing).	3 Hours	Ram
6	Write SQL query to get list of user's favorited properties	2 Hours	Ram
7	Create unit test to ensure that the SQL query works	2 Hours	Ram

Acceptance Criteria:

- Given that the like button is visible, when the user hovers over the button, they will see it toggled.
- Given that the like button is clickable, when the user clicks on the button, they will see the button change color.
- Given that the like button has been clicked, when the user refreshes the page, the button will still be the color that indicates that the property has been favorited.
- Given that a property has been favorited by a user, when the SQL query runs, a user can see a list of names of their favorite properties.

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

Task #	Description	Estimated Time	Owner
1	Write SQL query to get location of specific property.	2 Hours	Ram
2	Display map where location of the	4 Hours	Ram

	property is at.		
3	Display the address of property along with distance for how far the property is from a specific campus.	1 Hour	Ram
4	Write a unit test to ensure that the SQL query works.	2 Hours	Ram

Acceptance Criteria:

- Given the map is properly displayed, when the user views the property listing page, they will see the property pinned on the map indicating its location.
- Given that the map is clicked, when the user clicks on the pin on the map, they will see the distance the given property is from a specified campus.
- Given that the query is written correctly, when a user clicks on a location, they should still see the same property that matches information from the database.

Backlog:

User Registration

1. ~~As a user, I would like to be able to make an account by filling out a user registration form.~~
2. As a user, I would like to be able to login with my username & password.
3. ~~As a user, I would like my password to be hidden when I type it.~~
4. ~~As a user, I would like to verify my password before creating my account.~~
5. As a user, I would like to be able to change the university associated with my account to a different university.
6. As a user, I would like to be able to view and update my user bio.
7. As a user, I would like to change and reset my password.
8. As a potential tenant, I want the site to verify that listers are students by confirming their school email.
9. As a lister, I want the site to verify that any potential tenants are students by confirming their school email.

Home Page

10. ~~As a user, I would like to see a grid view of multiple properties in my selected area.~~
11. ~~As a user, I would like a map to show all properties available in my selected radius.~~
12. ~~As a user, I would view the name of the property, price, dates available for rent, and miles from campus of each property.~~
13. ~~As a potential tenant, I would like to see all the available property listings with images and their location, date range, and cost.~~
14. As a potential tenant, I would like to filter the properties based on the room of a house or an apartment.
15. As a potential tenant, I would like to filter the properties by specifying a given price range.
16. As a potential tenant, I would like to filter the properties by a certain number of beds and bathrooms.
17. ~~As a lister, I would like to have a button to create a new post.~~
18. ~~As a lister, I would like to access my inbox with a single button.~~
19. ~~As a potential tenant, I would like to have a button to view all of my favorite properties.~~

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

20. ~~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.~~
21. ~~As a potential tenant, I would like to have the option to favorite properties so I can view them later.~~
22. ~~As a potential tenant, I would like to be able to see the locations of the properties.~~
23. As a potential tenant, I would like to see the listers' contact information.
24. As a potential tenant, I would like to directly start a chat with the lister.
25. 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)

26. As a lister, I would like to be able to fill out a form with all the specific information about a property including paragraph description of the property and a list of amenities.

27. 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.
28. As a lister, I would like to be able to submit the information in the form and see it posted in the homepage listings.
29. As a lister, I would like to be able to view my posted listings on the page.
30. As a lister, I would like to be able to update my posted listings.
31. As a lister, I would like to see that the potential tenant is a verified student.

Message System

32. As a user, I would like to easily view all my messages.
33. As a potential tenant, I would like to be able to contact listers about their property.
34. As a lister, I would like to view all of my messages from potential tenants.
35. As a potential tenant, I would like to view all of my messages from listers.
36. As a lister, I would like to have my messages to other listers and my messages from potential tenants be separated.
37. As a potential tenant, I would like listers to have the names of their properties included to help differentiate them.

Map

38. As a potential tenant, I would like a map view to see where all of the properties are relative to campus.
39. As a lister, I would like the ability to input the location of my lease and have it show up on the map.
40. As a potential tenant, I would like the ability to scroll across the map and zoom in or out.
41. As a potential tenant, I would like the ability to search for an address on the map.
42. As a potential tenant, I would like my filters to be reflected onto the map.

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.