Sprint One Report

PORTFOLIO TASK 4

Unit code: COS40005
Unit Name: Engineering project A
Submission date: 18/10/2024
Team 5 Monday 10:30

Student Name	Student ID	Statement of contribution to the report
Ashim Adhikari	104104333	Planned sprint 4 to 5. Completed back end, connected GitHub repo with AWS services, AWS console creation and distributed user role
Aaradhya Lamsal	103828166	Modified design and implemented the website version of the app, and reviewed retrospect for sprint 3.
Ruffin Remad	103840173	Developed design and implementation as part of the sprint, reviewed key lessons learned and further improvements
Nur E Siam	103842784	Designed the layout, conducted user research, gathered feedback, performed color research, worked on the front-end and researched the real estate market in Bangladesh. For

		Sprint 3, as per the report worked on Sprint one review.
Shivam Sharma	103800575	Worked on sprint 3 quality plan and did the critical analysis of sprint 1 and also identified the testing phases and did sprint 3 documentation.
Ankit Malik	103531273	Completed the Product Backlog after consulting the team members. Worked on the prototype, implemented the search functionality and worked on the aesthetical layout of the listings page. Specifically (property cards).
Prabesh Bhattarai	104085535	Drafted sprint plan (sprint 1-3). Deployed AWS amplify on AWS to run site publicly.

ACKNOWLEDGMENT OF COUNTRY

Each team member identifies:

> the Traditional Owners of the land they lived on while completing this work (if living in Australia)…

1. PRODUCT BACKLOG

ID	Task/Feature	Description	Business Value (Priority)	Dependencies	Planned Sprint
1	Property Listings	Users can create property listings with details like property type, location, price, and images.	High	User Registration and Login	Sprint 1
2	Search and Filtering	Enable advanced search with filters for location, price, property type, and amenities.	High	Property Listings	Sprint 1
3	Interactive Map-based Search	Integrate Google Maps for proximity-based search to allow users to search properties by area or key landmarks.	High	Search and Filtering, Google Maps API	Sprint 1
4	User Registration and Login	Enable users to register, log in, and reset passwords using email/phone.	High	None	Sprint 2
5	Profile Management	Allow users to update personal details, including contact info, preferences, and professional verification status.	High	User Registration and Login	Sprint 2
6	Property Listing Management	Allow users to edit or delete their own listings and track listing performance.	Medium	Property Listings	Sprint 2
7	Messaging System	Implement secure in-app messaging for communication between users and real estate professionals.	High	User Registration and Login	Sprint 3

8	Notifications	Real-time notifications for new messages, updates on listings, or system alerts.	Medium	Messaging System, Property Listings	Sprint 3
9	Multi-language Support	Enable platform localization with support for Bengali and English.	Medium	UI/UX Design	Sprint 3
10	Secure Data Encryption	Implement data encryption for sensitive information (AES-256 and TLS).	High	None	Sprint 3
11	System for Verification Professionals	Implement multi-factor authentication and CAPTCHA to verify real estate professionals and ensure authenticity.	High	User Registration and Login	Sprint 4
12	User Reviews and Ratings	Allow users to rate and review properties and professionals after interactions.	Medium	Property Listings, Profile Management	Sprint 4
13	Image Upload Optimization	Automate image compression upon upload and optimize media storage using AWS S3.	Medium	Property Listings, AWS S3	Sprint 4
14	Appointment Scheduling	Implement a system for scheduling property visits and viewing times with reminders.	Medium	Messaging System, Property Listings	Sprint 4
15	Analytics and Reporting for Admin	Provide admin with analytics and reporting on user activities, property listings, and professional verifications.	Low	User Registration, Property Listings	Sprint 5
16	Admin Dashboard	Develop an admin dashboard for managing users, content moderation, and reporting.	Medium	Admin System, Analytics and Reporting	Sprint 5
17	Backup and Recovery	Implement automated backups and recovery procedures using AWS RDS and S3.	High	Database Setup (Amazon RDS)	Sprint 5

18	Scalability (Auto Scaling & Load Balancing)	Implement auto-scaling and load balancing to handle traffic spikes during peak periods.	High	AWS Setup, Backend Infrastructure	Sprint 5

2. QUALITY PLAN

The quality plan is essential to ensure that the real estate platform meets the **functional**, **performance**, **security**, and **user experience** requirements of the project. The goal is to deliver a high-quality product that addresses the specific needs of the Bangladeshi real estate market, ensuring user satisfaction and platform reliability. This plan outlines the **target quality expectations**, **quality assurance processes**, and the steps that will be taken to test and validate the system.

1. Target Quality Expectations

The following quality expectations have been set for the project:

- **Reliability**: The platform should have an uptime of at least 99.5%, ensuring that the system is available for users, especially during peak hours, such as the start of the academic year when students search for rental properties.
- **Performance**: The system should maintain fast load times (under 2 seconds) for key features like property search and messaging, even under high traffic (up to 10,000 concurrent users).
- **Security**: Data security is critical due to sensitive information (user profiles, professional verification, property listings). The platform must comply with industry-standard encryption (AES-256 and TLS) and undergo regular vulnerability testing to protect against security threats.
- **Scalability**: The system should scale dynamically to handle traffic spikes, ensuring seamless performance without downtime.
- **User Experience**: The platform should be easy to navigate, with intuitive search and filtering functions, multi-language support (Bengali and English), and responsive design for both web and mobile interfaces.
- **Functionality**: All core functionalities (e.g., user registration, property listings, messaging, and notifications) should work seamlessly, and user inputs must be validated effectively to prevent errors or incorrect data entry.
- **Accuracy and Trust**: The platform should implement a robust verification system for real estate professionals, ensuring that users trust the information provided.

2. Quality Assurance Process

To meet the quality expectations outlined above, the following steps will be taken to assure quality throughout the project:

2.1 Code Quality and Best Practices

- **Code Reviews**: Each piece of code will undergo **peer reviews** to ensure it adheres to best practices, such as proper use of functions, modularity, and clear naming conventions.
- **Linting and Formatting**: Tools like **ESLint** for JavaScript and **Prettier** for formatting will be used to maintain a consistent codebase and identify any potential issues early.
- **Version Control**: **Git** will be used to manage version control, with code changes pushed through a pull request process. This will ensure that all changes are reviewed and approved before being merged into the main branch.

2.2 Automated Testing

- **Unit Testing**: Automated unit tests will be written using frameworks like **Jest** for React.js and Node.js. Unit tests will cover critical components such as user registration, property listings, search functionality, and messaging.
    - **Objective**: Ensure that each component behaves as expected under different conditions.
    - **Target**: Achieve **90% test coverage** of the codebase.
- **Integration Testing**: Integration tests will verify that different components of the platform work well together, such as ensuring property listings integrate properly with the search and messaging systems. Tools like **Postman** and **Cypress** will be used for API and frontend integration tests.
    - **Objective**: Confirm that various parts of the system (e.g., backend and frontend) interact as expected.
- **End-to-End (E2E) Testing**: E2E tests will simulate real-world user scenarios, testing entire workflows such as registering, logging in, searching for properties, and messaging an agent. Tools like **Selenium** or **Cypress** will be used for this purpose.
    - **Objective**: Validate the full user journey to ensure a seamless experience.
    - **Target**: Ensure critical user journeys (such as property search, messaging, and login) are fully functional.

2.3 Manual Testing

- **User Acceptance Testing (UAT)**: Before deployment, **user acceptance testing** will be conducted to verify that the platform meets user expectations. Real users (students,

property buyers, agents) will test key features to ensure usability, functionality, and trust.
    - **Objective**: Ensure the platform meets the needs of the Bangladeshi real estate market and delivers a good user experience.
    - **Feedback**: User feedback from UAT will be analyzed, and necessary changes will be implemented in the next sprint.

2.4 Security Testing

- **Penetration Testing**: External penetration testing will be performed to identify and address any vulnerabilities, such as SQL injection or cross-site scripting (XSS).
- **Data Encryption Validation**: Tests will verify that sensitive information (passwords, personal details) is encrypted using **AES-256** and **TLS** protocols.
    - **Objective**: Ensure data security and compliance with privacy regulations.

2.5 Performance and Load Testing

- **Load Testing**: Tools like **Apache JMeter** or **Loader.io** will be used to simulate high traffic (up to 10,000 concurrent users) to ensure the platform can handle peak loads.
    - **Objective**: Validate system performance and ensure response times remain under 2 seconds for 95% of the transactions.
- **Scalability Testing**: AWS **auto-scaling** features will be tested to ensure that additional resources are provisioned automatically during traffic spikes.
    - **Objective**: Ensure the platform can dynamically scale to handle increased user demand.

2.6 UI/UX Testing

- **Usability Testing**: Regular usability tests will be conducted to assess the user experience, ensuring that navigation is intuitive and responsive. Tools like **Hotjar** or **UserTesting.com** will help monitor user interactions and feedback.
    - **Objective**: Ensure the platform is easy to use and navigate, providing a positive experience for both tech-savvy and non-tech-savvy users.
    - **Multi-Language Testing**: The platform's localization (Bengali and English) will be tested to ensure all text is displayed correctly and contextually in both languages.

3. Planned Testing Stages

Testing will occur throughout the development process, integrated into each sprint cycle to ensure that bugs are caught early, and features are validated as they are developed.

3.1 Sprint 1: Basic Features Testing

- **Unit Testing**: Focus on user registration, login, and profile management.
- **Integration Testing**: Ensure registration connects properly with user profiles.

3.2 Sprint 2: Search and Map Testing

- Unit and Integration Testing: Validate property listings and search functionality.
- End-to-End Testing: Test the interactive map and filtering features with real-time data.

3.3 Sprint 3: Messaging and Notification Testing

- Unit and Integration Testing: Ensure real-time messaging and notifications are working across different user roles (buyers, sellers, agents).
- Performance Testing: Begin load testing with moderate traffic levels (up to 5,000 concurrent users).

3.4 Sprint 4: Security and Verification Testing

- Security Testing: Focus on penetration tests, CAPTCHA, and multi-factor authentication.
- Data Encryption Testing: Verify that sensitive information is properly encrypted.

3.5 Sprint 5: Full System Testing

- **End-to-End Testing**: Simulate complete user journeys across all features, ensuring all components (messaging, search, registration) work seamlessly together.
- **Performance and Load Testing**: Test the platform's scalability with up to 10,000 concurrent users.

4. Continuous Integration (CI) and Continuous Deployment (CD)

The project will use a **CI/CD pipeline** to ensure that code changes are automatically tested and deployed. **GitHub Actions** or **Jenkins** will be used to automate the following processes:

- **Code Integration**: Automatically run unit and integration tests on every commit or pull request.
- **Automated Deployment**: Successful builds will be deployed to the testing environment for further evaluation and feedback.

3. SPRINT PLAN

1. **Sprint 1: Initial Design, Architecture Setup, and Prototyping**
   Timeline: Semester 1, Weeks 9-11
   Goal: Create a solid foundation with architecture and initial prototypes for the real estate application.
   a) **Week 9**:
      - Finalize project roadmap.
      - Develpment of front-end page with buy, rent, and sold, pages.
      - Add fully working search functions to the webpage for searching properties.

   b) **Week 10**:
      - Set up AWS backend environment (AWS Amplify).
      - Deploy the code in AWS cloud and test the connectivity.

   c) **Week 11**:
      - Finalize basic UI prototypes for web application with at least 1-2 function working
      - Add a basic google map and pinpoint the location of the properties in the map.
      - Integrate the frontend prototype with AWS backend to access the page public form any geographical region with internet access.

   Deliverables: Project roadmap, architecture, basic UI prototype, AWS environment setup.

2. **Sprint 2: Backend Development and Core Functionality Implementation**
   Timeline: Semester 2, Weeks 1-3
   Goal: Implement core backend functionalities, focusing on security, user authentication, and media management.

   a) Week 1:
      - Finalize database schema using MySQL on AWS RDS.
      - Start backend logic with AWS services functions for property listings management.

   b) Week 2:
      - Set up secure user authentication using AWS Cognito.
      - Enable two-way multi-factor authentication using OTP code via email or phone while singing in.
      - Configure AWS S3 for media storage (images, videos), and connect it to the backend.

   c) Week 3:
      - Integrate Interactive map-based search for some popular cities of Bangladesh where many colleges and universities are located.
      - Start working on basic mobile application user interface and choosing appropriate colour scheme.

Deliverables: Database setup, media management, and database integration.

3. **Sprint 3: Real-Time Interaction, Frontend Enhancement and security enhancement**
   Goal: Enable real-time interaction, improve the frontend, and add location-based services like map integration.
   a) **Week 4**:
      - Implement real-time messaging using WebSocket for communication between users and professionals.
      - Continue frontend integration with ReactJS for web and mobile apps.

   b) **Week 5**:
      - Test real-time messaging and property search functionalities.
      - Enhance frontend for responsive design across different screen sizes and devices.
      - Make good user interface to view property details. This interface will show property features including amenities.

   c) **Week6**:
      - Implement ACLs for configuring basic security measure during development phase of sprint 3.
      - Configure basic settings for AWS WAF (Web Application Firewall) to analyse security settings.

   Deliverables: Real-time messaging, responsive and optimized frontend and security enhancement.

4. **Sprint 4: Security Implementation, Testing and Database Setup**
   Timeline: Semester 2, Week 7-9

   Goal: Implement security features, conduct initial testing, and set up the database infrastructure for the real estate application.

Week 7:

Integrate AWS WAF for web application security against SQL Injection and XSS attacks.

- Set up and configure Amazon Cognito for user authentication with MFA.
- Implement access control mechanisms for different user roles (e.g., Admin, Property Owner, User).

Week 8:

- Deploy Amazon RDS for database management and ensure high availability using multi-AZ deployment.
- Conduct security tests focusing on firewall rules, access control, and user authentication.
- Set up logging and monitoring services using Amazon CloudWatch for security and performance insights.

Week 9:

- Optimize network security by configuring subnets, security groups, and NACLs within the VPC.
- Perform security and performance benchmarking using automated testing tools.
- Review security implementation, conduct peer reviews, and document test cases and results.

Deliverables:

- Security features (AWS WAF, Cognito) integrated.
- RDS setup and tested.
- CloudWatch monitoring configured.
- Security tests completed and documented.

Sprint 5: Load Balancing, Scalability, and Final Testing

Timeline: Semester 2, Weeks 10-12

Goal: Ensure scalability, conduct final testing, and refine the application for deployment.

Week 10:

- Set up Amazon Route 53 for DNS routing with health checks for load balancing.
- Configure Amazon CloudFront for content delivery and performance optimization.
- Test load handling using AWS Elastic Load Balancer (ELB) across multiple Availability Zones.

Week 11:

- Implement NAT Gateway and Internet Gateway for secure internet access to instances.
- Optimize the caching mechanisms for faster content delivery using CloudFront.
- Conduct load testing to ensure the application can handle peak traffic without downtime.

Week 12:

- Complete the integration of SNS for event-driven notifications (e.g., new property listings, system alerts).
- Final round of end-to-end testing, covering performance, security, and scalability.
- Review and finalize documentation for all AWS services used and their configurations.

Deliverables:

Route 53 and ELB configured.
CloudFront and caching optimized.
Load testing completed.
Final testing report and system documentation ready for deployment.

4. SPRINT ONE REVIEW

Sprint 1 Plan (Deliverables Agreed with the Client/Supervisor)

The objective of Sprint 1 was to lay the groundwork for the platform's infrastructure and functionality, concentrating on both front-end and back-end components. The primary deliverables included:

- User Registration and Login: Develop a fundamental user registration and login process, incorporating form validation to ensure accurate data entry. Users should have the option to register using either their email or phone number and access their accounts seamlessly.
- Property Listings: Enable users to create property listings by providing key information such as property type, location, price, and images.
- Search and Filtering: Introduce a search feature equipped with filters to refine results based on property type, price range, and location.
- Interactive Map-Based Search: Incorporate Google Maps to facilitate property searches based on geographical location, represented through map markers.
- Backend Setup Using AWS Amplify: Establish AWS Amplify to manage the backend infrastructure, user authentication, and data handling. This setup will serve as the foundation for the platform's scalability and security.

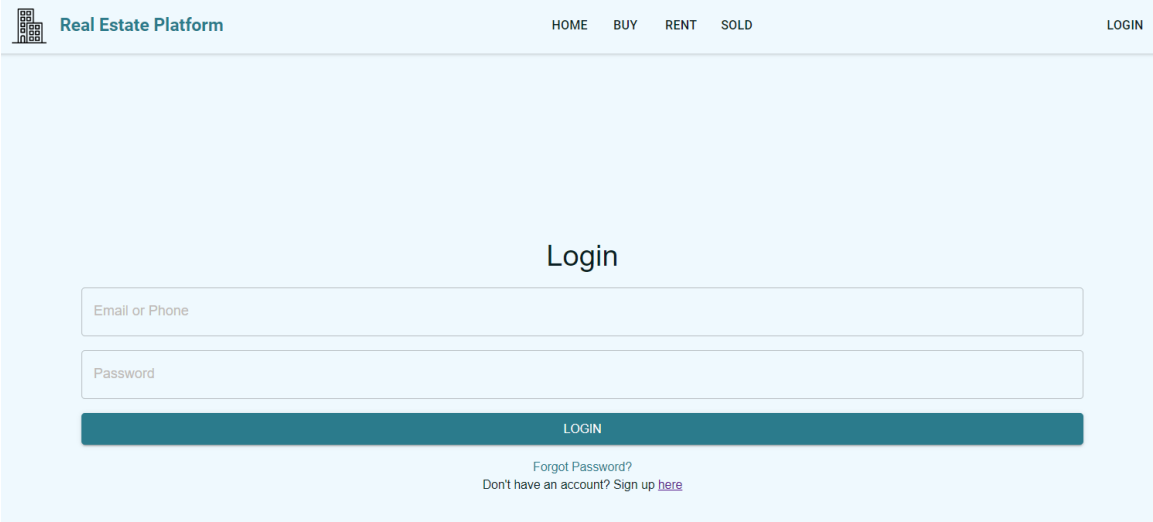
The goal was to ensure that at least 2-3 of these features were operational by the conclusion of the sprint, with additional enhancements planned for subsequent sprints.

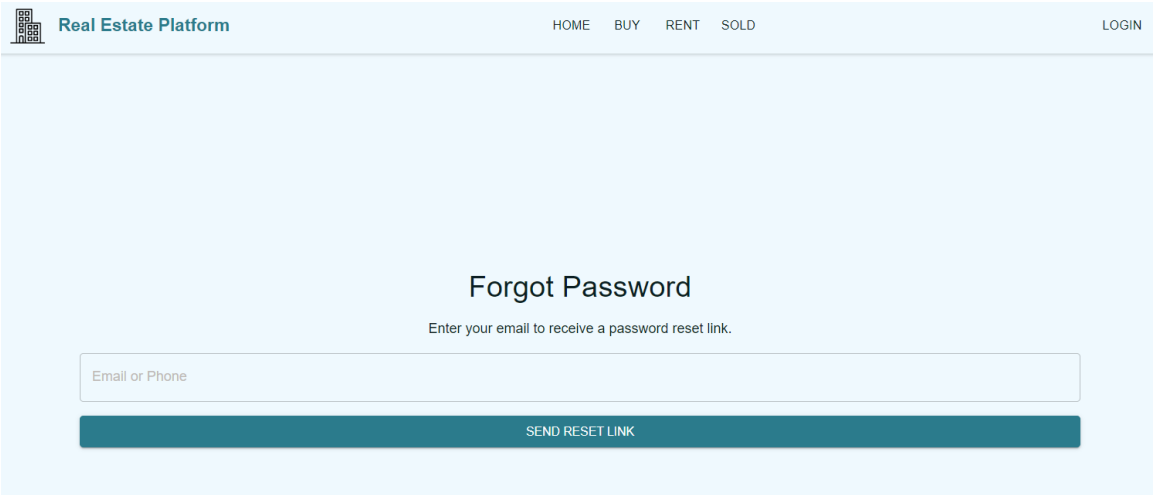
Progress Made in the First Sprint (Description with Screenshots)

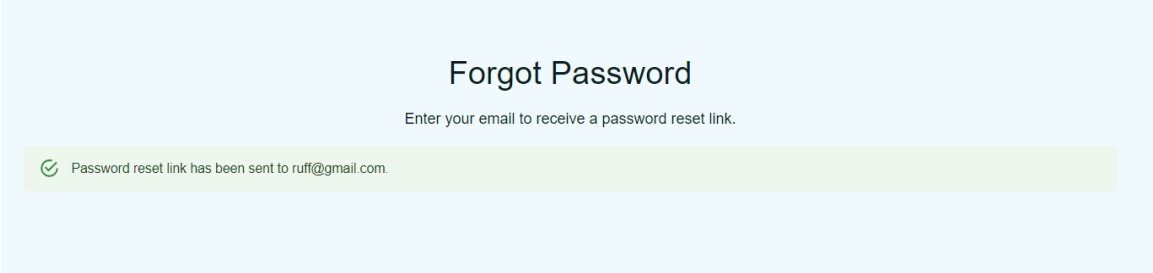
The team achieved significant milestones during Sprint 1, exceeding the initial objectives by delivering five functional components. Below is a summary of the primary advancements:

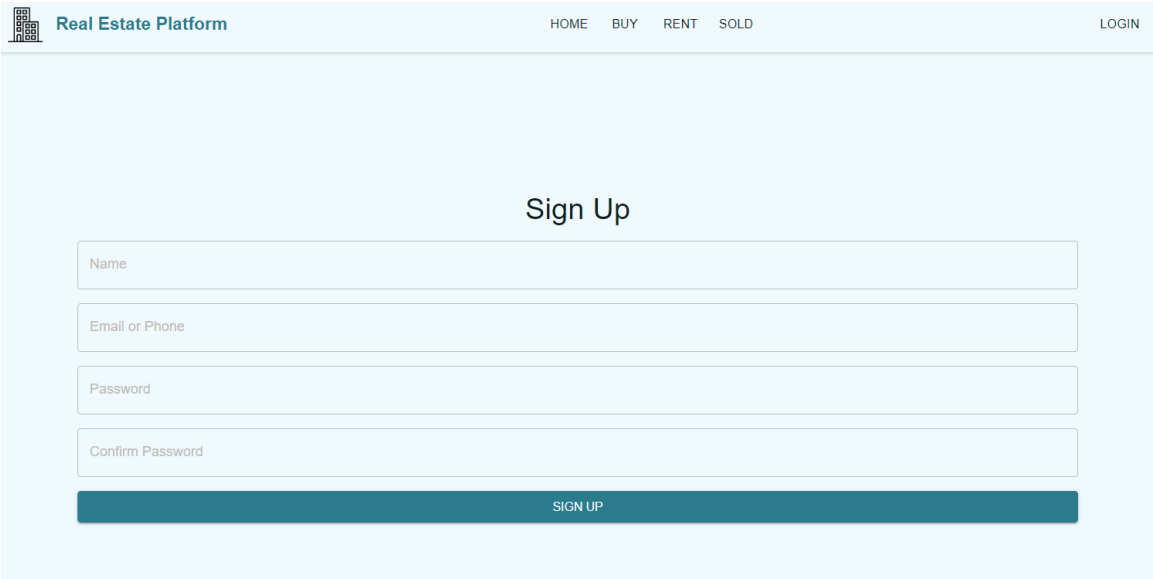
User Registration and Login:

The team successfully created a fully functional registration form that includes validation for both email addresses and phone numbers. While the front-end aspect is complete, the backend integration for user data storage is still in progress. The login form is also operational on the front end, but similar to the registration process, the backend implementation is yet to be finalized.



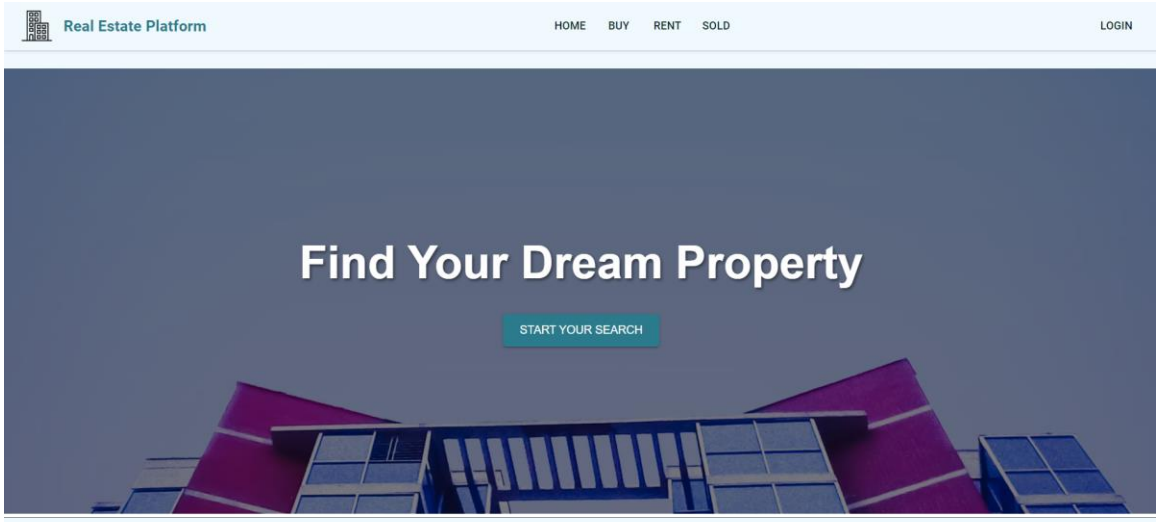


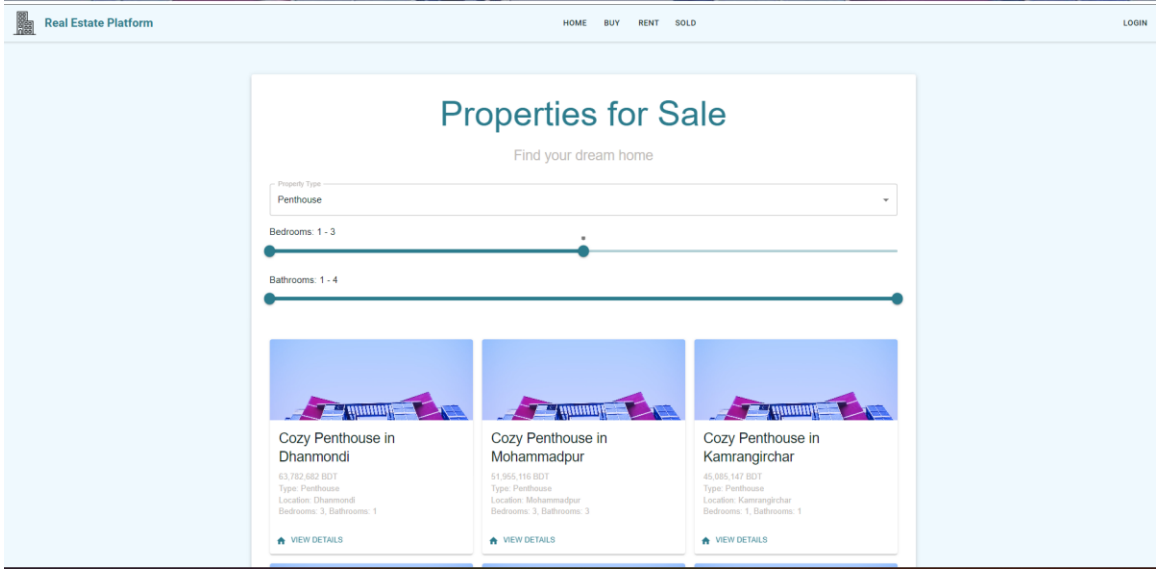


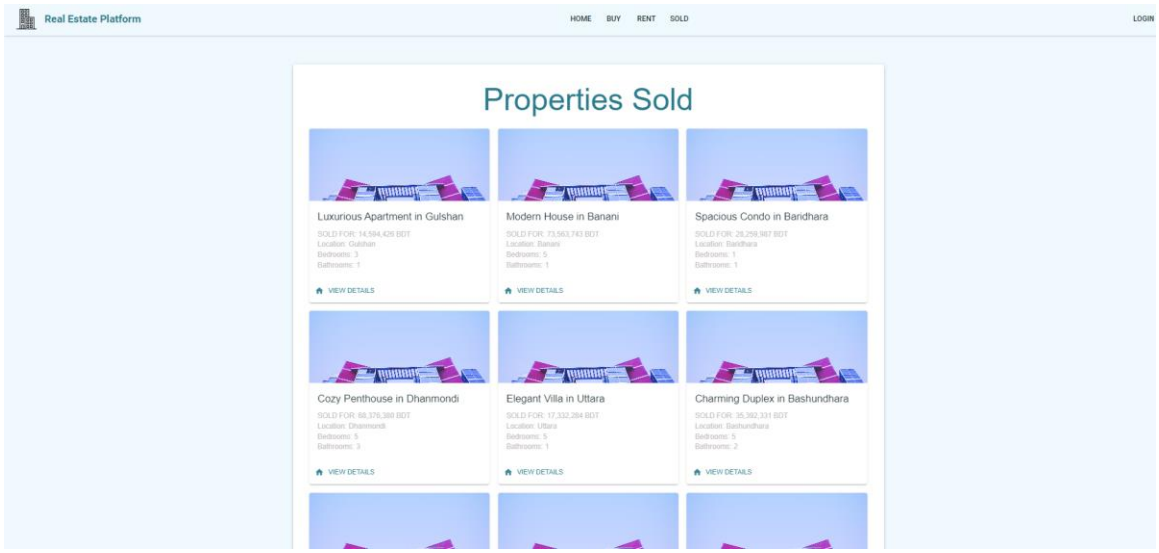


Property Listings:

A front-end form has been developed that allows users to create property listings. This form enables users to enter key information such as property type, location, price, and the number of bedrooms and bathrooms. Although the form is fully operational, the backend functionality for saving and retrieving these listings is still under construction.

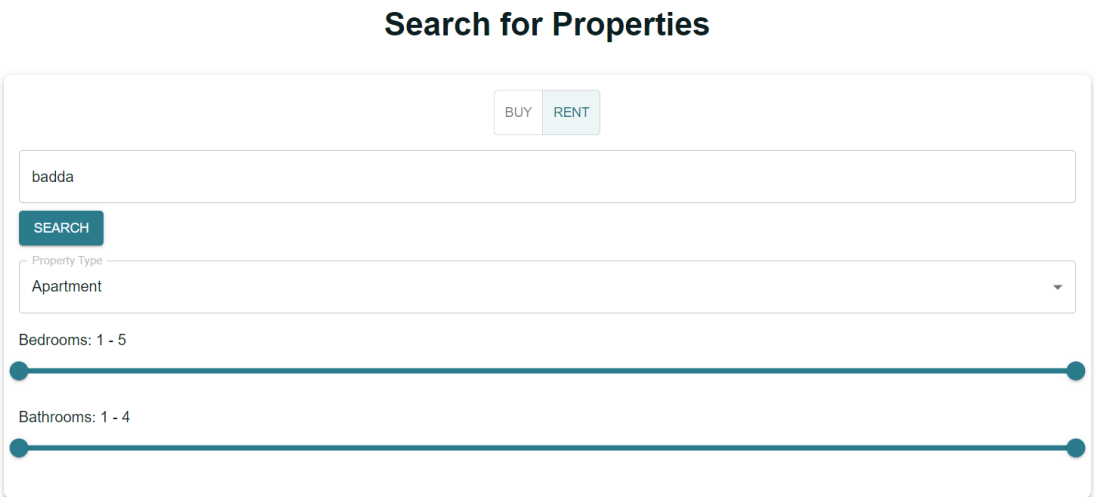

Real Estate Platform
HOME
BUY
RENT
SOLD
LOGIN
Find Your Dream Property
START YOUR SEARCH


Real Estate Platform
HOME
BUY
RENT
SOLD
LOGIN
Properties for Sale
Find your dream home
Property Type
Penthouse
Bedrooms: 1 - 3
Bathrooms: 1 - 4
Cozy Penthouse in Dhanmondi
63,782,682 BDT
Type: Penthouse
Location: Dhanmondi
Bedrooms: 3, Bathrooms: 1
VIEW DETAILS
Cozy Penthouse in Mohammadpur
51,955,116 BDT
Type: Penthouse
Location: Mohammadpur
Bedrooms: 3, Bathrooms: 3
VIEW DETAILS
Cozy Penthouse in Kamrangirchar
45,085,147 BDT
Type: Penthouse
Location: Kamrangirchar
Bedrooms: 1, Bathrooms: 1
VIEW DETAILS


Real Estate Platform
HOME
BUY
RENT
SOLD
LOGIN
Properties Sold
Luxurious Apartment in Gulshan
Modern House in Banani
Spacious Condo in Baridhara
Cozy Penthouse in Dhanmondi
Elegant Villa in Uttara
Charming Duplex in Bashundhara
VIEW DETAILS

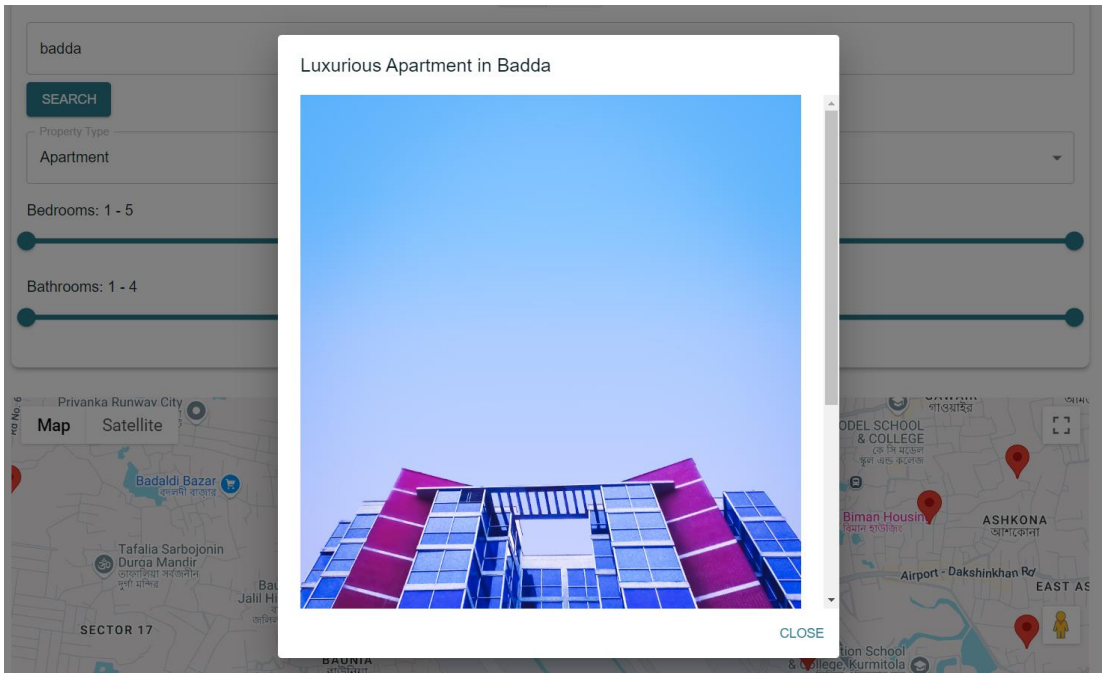
Search and Filtering:

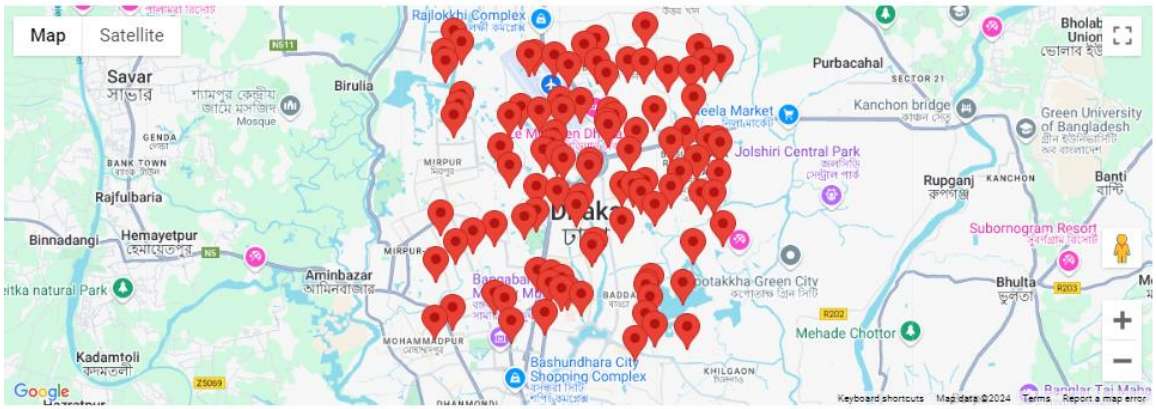
The implementation of the search bar and filtering options has been successfully completed. Users can search for properties based on location and property type, and utilize sliders to filter results by the number of bedrooms and bathrooms. This search feature is integrated with Google Maps, offering users a dynamic, location-based property search experience.



Interactive Map-Based Search:

A notable achievement in Sprint 1 was the successful integration of Google Maps. Properties are displayed as red markers throughout Dhaka, allowing users to visually search for available listings in their preferred areas. This feature greatly enhances the platform's usability, providing users with a real-time geographic overview of properties.

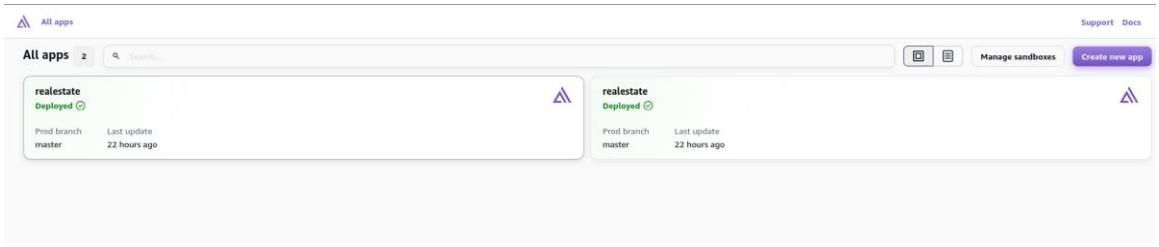


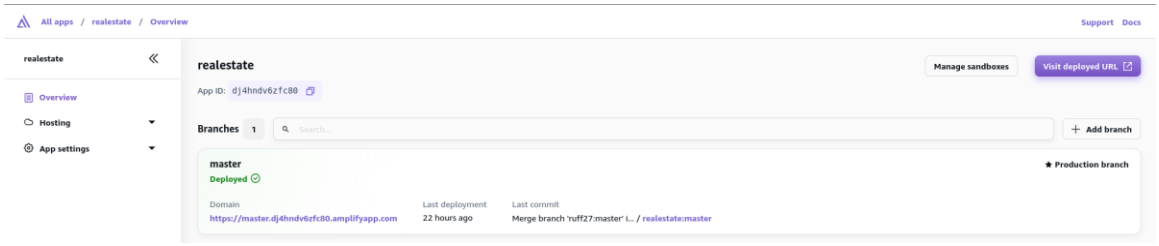


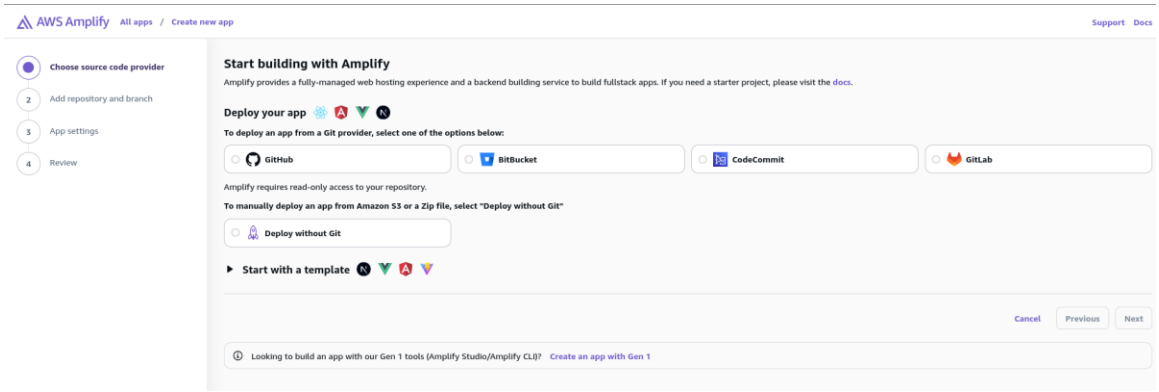
Backend Setup via AWS Amplify:

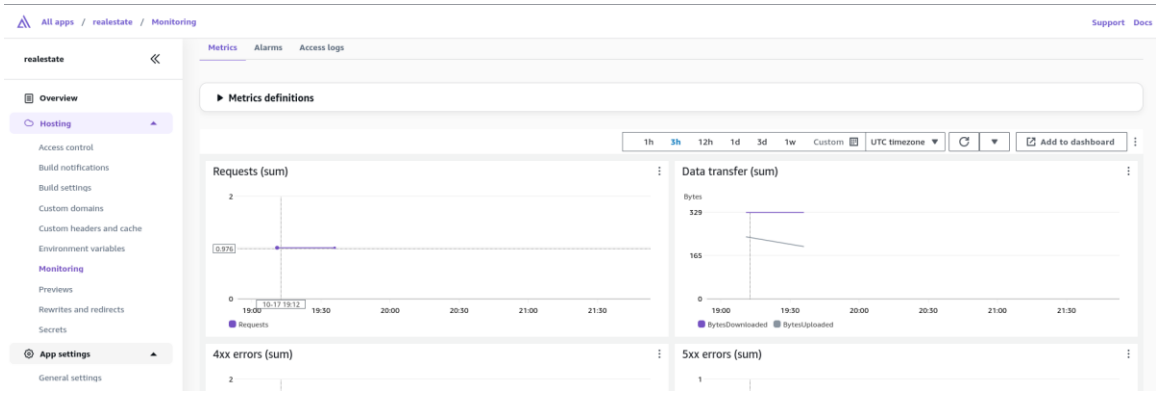
The team effectively integrated the project with AWS Amplify to establish the backend infrastructure. Although there were initial challenges related to AWS security tokens that hindered progress, the team resolved these issues by correctly configuring the AWS credentials, allowing them to successfully deploy everything to the cloud.

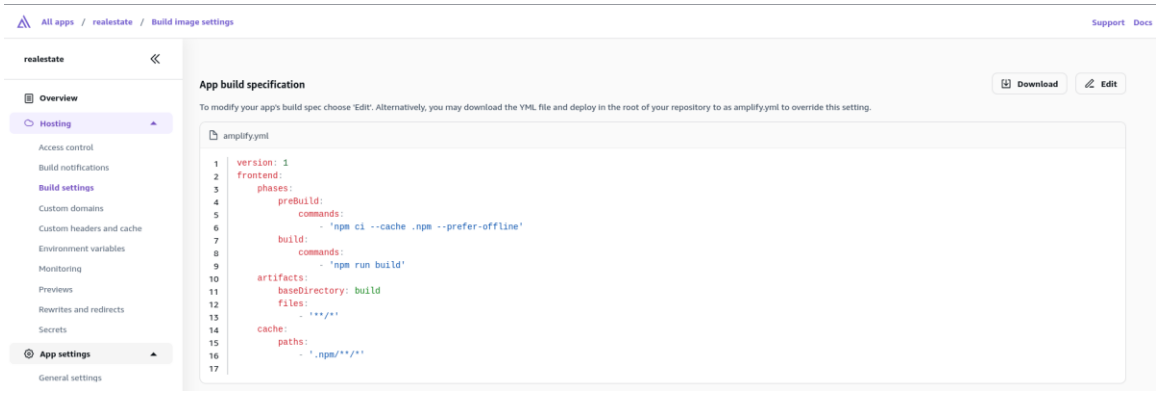
To manage access, IAM users were established, guaranteeing that only authorized services and users can access the AWS resources. Additionally, monitoring tools and dashboards were implemented to track requests, data transfers, and errors, offering valuable insights into backend performance.











Overall, this configuration guarantees a scalable and secure backend, providing a robust foundation for future advancements.

Feedback Received from the Client on the Deliverable

The client expressed satisfaction with the advancements achieved during Sprint 1. The deliverables aligned with initial expectations, and the client provided the following suggestions for enhancement:

1. Default Rent Option: The client appreciated the property search and filtering features but requested that the "Rent" option be set as the default when the web application launches, instead of the current "Buy" option. This minor change would better reflect user behavior, as renting is typically the more common initial action.

2. Sold Properties Page with Statistics: The client requested the creation of a “Sold” page that would showcase statistics related to sold properties, such as the average selling price by area and pricing trends across different regions. Although this was not included in the scope of Sprint 1, it will be prioritized in upcoming sprints.

3. Enhancements to Login and Sign-Up: While the registration and login forms on the front end are operational, the client highlighted the necessity for back-end functionality to ensure these features work effectively. This will be addressed as a priority in the next sprint.

In summary, the client was pleased with the progress made but recommended several minor adjustments to enhance the overall user experience.

A Critical Analysis of Progress Against the Plans

The outcomes of Sprint 1 surpassed our original expectations, with the team successfully delivering a greater number of features than planned. Below is a summary of the key achievements and challenges encountered:

Achievements:

- Front-end development: The team finalized the front-end functionalities for user registration, property listings, search and filtering, as well as the interactive map. These components are fully functional from a user interface standpoint and are prepared for back-end integration.
- Google Maps integration: A significant milestone of Sprint 1 was the integration of Google Maps, which provides users with an intuitive method for searching properties based on location.
- AWS Amplify setup: The back-end of the project was effectively connected to AWS Amplify, ensuring scalability, security, and readiness for future data management requirements. IAM roles were also created to facilitate controlled access to AWS resources.

Challenges:

- AWS Integration: The team faced some difficulties during the AWS Amplify setup, particularly concerning the configuration of security tokens. This challenge caused delays in the back-end setup but was ultimately resolved.
- Back-end functionality: Although the front-end is fully developed for essential features, the back-end functionalities for user registration, login, and property listings are still pending completion. This will be prioritized in Sprint 2 to facilitate secure data storage, retrieval, and user authentication.

Next Steps:

- Back-end integration: Finalize the back-end setup for user registration, login, and property listings to allow users to securely store and retrieve data.
- Implement the "Sold" page: In response to the client’s request, this page will showcase property sale statistics and offer insights into the real estate market.
- Adjust the Rent default option: Modify the web application to set the "Rent" option as the default, as per the client’s specifications.

In conclusion, Sprint 1 was a notable success. The team established a robust foundation for the platform and delivered more functionality than initially anticipated. With the client's feedback, we are well-equipped to maintain this momentum in upcoming sprints.

5. RETROSPECT

Using an organized methodology, our team broke the project up into three primary sprints, each with a prep and wrap-up phase. Our objectives could be regularly reevaluated and developed iteratively thanks to this framework.

Key aspects of our process:

1) Weekly gatherings to talk about achievements and difficulties
2) Using SharePoint for document management and Microsoft Teams for collaboration
3) Precise descriptions of roles based on personal strengths
4) Continual revisions to user stories and project backlogs

Throughout the sprints, we encountered several challenges:

1) Part-time Availability: Different schedules among team members made collaboration challenging.
   Solution: Expanded the usage of asynchronous communication tools and instituted flexible meeting schedules.

2) Skills Gaps: A few team members lacked project-related knowledge of technologies.
   Solution: Set up peer-learning meetings and made use of internet tools to improve skills.

3) Technical Complexity: A few project features required sophisticated implementations.
   Solution: Allotted more time for extensive study and feature prototyping.

4) Project Scope Management: It was difficult to strike a balance between the deadlines for features and development.
   Solution: To efficiently manage jobs and modify scope as needed, a priority matrix was implemented.

Sprint-specific Observations

SPRINT 1

1) Focus on requirement engineering and initial development
2) Challenge: Aligning on technical architecture decisions
3) Outcome: Successfully established our core infrastructure

SPRINT 2

1) Emphasis on user-centered design and core feature development
2) Bottleneck: Integrating third-party APIs with our services
3) Resolution: Conducted focused technical spikes to resolve integration issues

SPRINT 3

1) Concentrated on risk management, quality assurance, and communication
2) Challenge: Balancing bug fixes with new feature development
3) Approach: Implemented a priority matrix to manage tasks effectively

Key Learnings and Improvements

1) **Documentation**: Improved our documentation process for better team alignment.
2) **Task Prioritization**: Developed a more effective system for prioritizing and managing tasks.
3) **Communication**: Enhanced our communication protocols to ensure all team members stayed informed.
4) **Retrospectives**: Started conducting end-of-sprint retrospectives to continuously improve our process.

Though we confronted and overcame substantial hurdles relating to part-time availability, talent inequalities, and project complexity, our organized approach gave us a strong foundation for development. As the project progressed, our capacity to modify our procedures was essential, especially in the areas of task management and information exchange. Going forward, we want to increase communication, continue to cultivate a culture of continuous learning and improvement within the team, and further refine our development techniques.

6. LESSONS LEARNED

1. Critical review of Sprint 1 experience:

- **Milestones**
  - **Foundations Laid:** The core architecture was established using AWS services, ensuring a scalable backend infrastructure. The frontend was developed with key pages, including home, rent, and sold property pages. Additionally, deploying the site on the cloud through AWS Amplify allowed public access, setting the stage for further development and testing.
  - **Team Collaboration:** The team effectively used Microsoft Teams for communication and SharePoint for document management, ensuring smooth coordination. Weekly

meetings allowed members to track progress and address challenges, while tasks were assigned based on individual strengths. Although communication was generally effective, better alignment on task dependencies will be prioritized in future sprints.

- **Progress of Implementation:** Significant progress was made on the frontend, including the implementation of search and filtering features. The Google Maps API was integrated, enabling users to search properties by location. However, backend delays prevented some features, like user registration, from being fully functional during this sprint.

- **Challenges**
  - **Technical Complexities:** Integrating AWS services and the Google Maps API was more challenging than anticipated, leading to some delays. Configuring authentication and ensuring the seamless connection between frontend and backend components required additional effort. These issues highlighted the need for earlier technical planning and testing.
  - **Time Management:** Conflicting schedules among team members made it challenging to align tasks effectively. Delays in backend development affected dependent features, slowing overall progress. Moving forward, better time management strategies, including parallel development of frontend and backend, will be implemented.
  - **Learning Curves:** Team members faced challenges in working with new tools, including AWS Amplify and Google Maps API, which initially slowed development. Peer learning sessions and external resources were used to build competence with these technologies. This experience reinforced the importance of continuous learning and skill sharing within the team.

- **Lessons Learned**
  - **Early Risk Management:** Identifying and mitigating potential risks early is critical, especially when working with external APIs and cloud infrastructure. A more comprehensive risk assessment during the initial planning phase could have minimized delays. Moving forward, the team will conduct detailed risk evaluations at the start of each sprint to anticipate challenges and develop mitigation strategies.
  - **Importance of Agile Adaptation:** The ability to adapt to changing priorities and unforeseen challenges proved essential during this sprint. Agile practices, such as reprioritizing tasks and reallocating resources, enabled the team to stay on track despite delays. This flexibility will remain a core part of the team's approach in future sprints to ensure continued progress.
  - **Continuous Communication:** While the team used effective communication tools, occasional gaps impacted task coordination. More frequent check-ins, particularly during critical phases, will be implemented to prevent misalignment. Moving

forward, the team will enhance both synchronous and asynchronous communication to ensure that everyone stays informed and aligned.

2. Future Plan

- **Goals for Sprint 2**
  - **Core Feature Development:** Focus on completing backend development using MySQL on AWS RDS to store user data and property listings. This will ensure smooth data management and allow for essential functionalities like property listings, search, and user profiles to work seamlessly.
  - **Backend database:** Develop secure backend logic using AWS Cognito for authentication with multi-factor authentication (MFA) to enhance security. Integration of user registration, login, and profile management will be a priority, ensuring that the platform is ready for user onboarding.
  - **Mobile app development:** Begin the design and development of the mobile version of the platform using React Native or a similar framework. The goal is to ensure the mobile app mirrors the web version's features, providing a consistent and intuitive experience for users across devices.

- **Improvements to Development Process**
  - **Enhanced Sprint Planning:** Break down tasks into smaller, actionable components to improve task tracking and completion. Allocate buffer time for complex tasks involving API integrations or cloud services to handle unexpected challenges efficiently. Use continuous integration pipelines to streamline deployment and testing processes, reducing the risk of last-minute issues.
  - **Task Management and Communication:** Implement more frequent check-ins and status updates to prevent miscommunication or delays. Use project management tools like Trello or Jira to ensure all tasks are assigned, tracked, and updated in real-time.

- **Long-term Outlook**
  - **Scaling for Future Sprints:** Set up AWS auto-scaling and load balancing to handle traffic surges and ensure system reliability under varying loads. Use AWS CloudWatch for performance monitoring, providing real-time insights into system health and enabling proactive issue resolution.

- **User Feedback:** Collect feedback from early users to identify pain points and improve features such as property search and Google Maps integration. Use this feedback to refine the user experience and ensure the platform meets the needs of the Bangladeshi real estate market effectively.