

Design Plan

PORTFOLIO TASK 3

Unit code: COS40005 EAT40005

Unit Name: Computing Technology/ Engineering project A

Submission date:

Student Name	Student Id	Statement of contribution to the report
Prabesh Bhattarai	104085535	System architecture and system design research
Nur E Siam	103842784	Solution approach, Other research (Survey and color scheme), Research and investigations
Shivam Sharma	103800575	Research on technical platform
Ashim Adhikari	104104333	Alternative Architecture, amazon compute service research
Aaradhya Lamsal	103828166	Research and investigations
Ruffin Remad	103840173	Problem overview, prototype design, KOST analysis
Ankit Malik	103531273	Research into the application domain

ACKNOWLEDGMENT OF COUNTRY

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campus is in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations. Each team member identifies the Traditional Owners of the land they lived on while completing this work (if living in Australia).

1. PROBLEM OVERVIEW

The real estate market in Bangladesh faces inefficiencies due to the lack of a centralized platform with features like real-time updates, interactive maps, and user verification. To address this, this project is committed to developing a mobile and web platform using **Amazon AWS** for scalable cloud hosting, **ReactJS** for the web frontend, **MySQL** for managing property listings and user data, and **Kotlin** for Android mobile development. The team brings experience in handling projects with **ReactJS**, **MySQL**, and **AWS**, and has conducted research on the advantages of these technologies, particularly their scalability, security, and cost-effectiveness. AWS ensures reliable cloud infrastructure, **ReactJS** allows for dynamic and responsive user interfaces, and **MySQL** offers robust data management. The platform's goal is to enhance property transactions by integrating real-time map-based search and secure user verification, addressing the limitations of current platforms like *Tolet* and *BD Housing*.

2. SOLUTION APPROACH

The solution architecture for the real estate platform is strategically designed using Amazon Web Services (AWS) to ensure scalability, security, and high performance for both mobile and web applications. The architecture enables seamless interaction between users and real estate professionals while maintaining data integrity and system reliability. Below is an overview of the high-level architecture, the components selected, and a brief discussion of alternatives that were considered but not chosen.

High-Level Architecture Design

The architecture is divided into distinct layers to ensure modularity, maintainability, and scalability:

1. User Interface Layer:

- **Mobile Application (Android):** The mobile app will be built for Android users, offering them functionality to browse property listings, communicate with professionals, and manage their profiles. The app will communicate with the backend via secure API endpoints exposed through AWS API Gateway.
- **Web Application:** The web app will offer a similar user experience to the mobile app, accessible via standard web browsers. Both applications will adhere to responsive design principles to ensure compatibility across different devices and screen sizes.

2. Backend Layer:

- **AWS Lambda (Serverless Architecture):** The core backend logic will be implemented using AWS Lambda, enabling a serverless approach that dynamically scales based on demand. Lambda will handle critical operations such as user authentication, managing property listings, handling API requests, and facilitating real-time communication between users.
- **Amazon RDS (MySQL):** A highly available, managed relational database will be used to store user profiles, property listings, transaction records, and other critical data. Amazon RDS provides built-in support for automatic backups, failover, and replication, ensuring data security and availability.
- **Amazon S3:** Media assets such as property images and other multimedia files will be securely stored in Amazon S3 buckets. This service offers highly scalable storage with lifecycle management policies to optimize storage costs. AWS Lambda functions will process and optimize images before they are uploaded to S3 for better performance and user experience.
- **AWS API Gateway:** API Gateway will serve as the entry point for all communication between the frontend applications (mobile and web) and the backend services, ensuring secure and scalable API interactions.

3. Security and Compliance Layer:

- **Multi-Factor Authentication (MFA) and CAPTCHA** will be implemented to enhance security during user authentication, preventing unauthorized access and ensuring data integrity.
- **End-to-End Encryption:** All communications between clients (mobile/web) and the server will be secured using HTTPS and TLS protocols, ensuring that sensitive data such as user credentials and property details are encrypted both in transit and at rest.
- **AWS Identity and Access Management (IAM):** Strict access controls will be enforced using IAM roles and security groups, ensuring that only authorized users and services can access sensitive resources.

4. Real-Time Interaction Layer:

- Google Maps API: Integrated for location-based property searches, the Google Maps API will allow users to view property listings on an interactive map, providing a more intuitive and efficient search experience.
- AWS SES (Simple Email Service): AWS SES will be used to send email notifications, alerts, and communication messages to users, such as notifications about new property listings or messages from real estate professionals.
- Web Sockets: Web Sockets will be used to facilitate real-time communication between users and real estate professionals, ensuring fast and efficient message delivery without requiring page reloads.

5. Scalability and Reliability Layer:

- Auto Scaling and Load Balancing: The architecture leverages AWS Auto Scaling to automatically adjust the number of EC2 instances in response to traffic demand, ensuring high availability and performance. An Elastic Load Balancer will distribute incoming traffic evenly across multiple instances, preventing any single point of failure.
- Amazon RDS Replication: RDS will utilize multi-AZ (Availability Zone) replication to ensure data redundancy and high availability. In case of a failure in one availability zone, the secondary instance in a different zone will take over, minimizing downtime.
- Amazon CloudFront: AWS CloudFront, a content delivery network (CDN), will be used to cache and deliver content globally with low latency. This will improve the performance of media-rich content such as property images and videos.

Architecture Alternatives Considered but Not Chosen:

1. Monolithic Architecture: A monolithic architecture was initially considered due to its simplicity and ease of implementation. However, it was ultimately rejected due to the limitations it imposes on scalability, flexibility, and the ability to support future growth. In contrast, the chosen microservices and serverless architecture using AWS Lambda offers greater scalability and better resource utilization, enabling the platform to handle high traffic efficiently.
2. On-Premises Infrastructure: Hosting the platform on an on-premises infrastructure was evaluated but discarded due to high upfront costs, maintenance overhead, and the lack of elastic scalability. AWS cloud infrastructure provides a more cost-effective solution, with the ability to scale resources dynamically based on demand, reducing both operational complexity and capital expenses.
3. Third-Party Hosting Solutions: While third-party managed services such as Digital Ocean and Heroku were briefly considered, they were ultimately rejected due to limitations in control, scalability, and integration with advanced AWS services like RDS, S3, and CloudFront. AWS offers a comprehensive suite of services that integrates seamlessly, providing the necessary tools to support both current and future platform requirements.

The selected architecture harnesses the power of AWS cloud services, ensuring that the real estate platform is highly scalable, secure, and capable of handling large volumes of traffic and data. The modular design enables seamless integration of future features, such as enhanced verification processes and advanced analytics, while the serverless and auto-scaling infrastructure ensures cost efficiency. This robust architecture positions the platform to meet the evolving needs of the Bangladeshi real estate market and offers users a modern, efficient, and secure platform for property transactions.

2.1 SYSTEM ARCHITECTURE

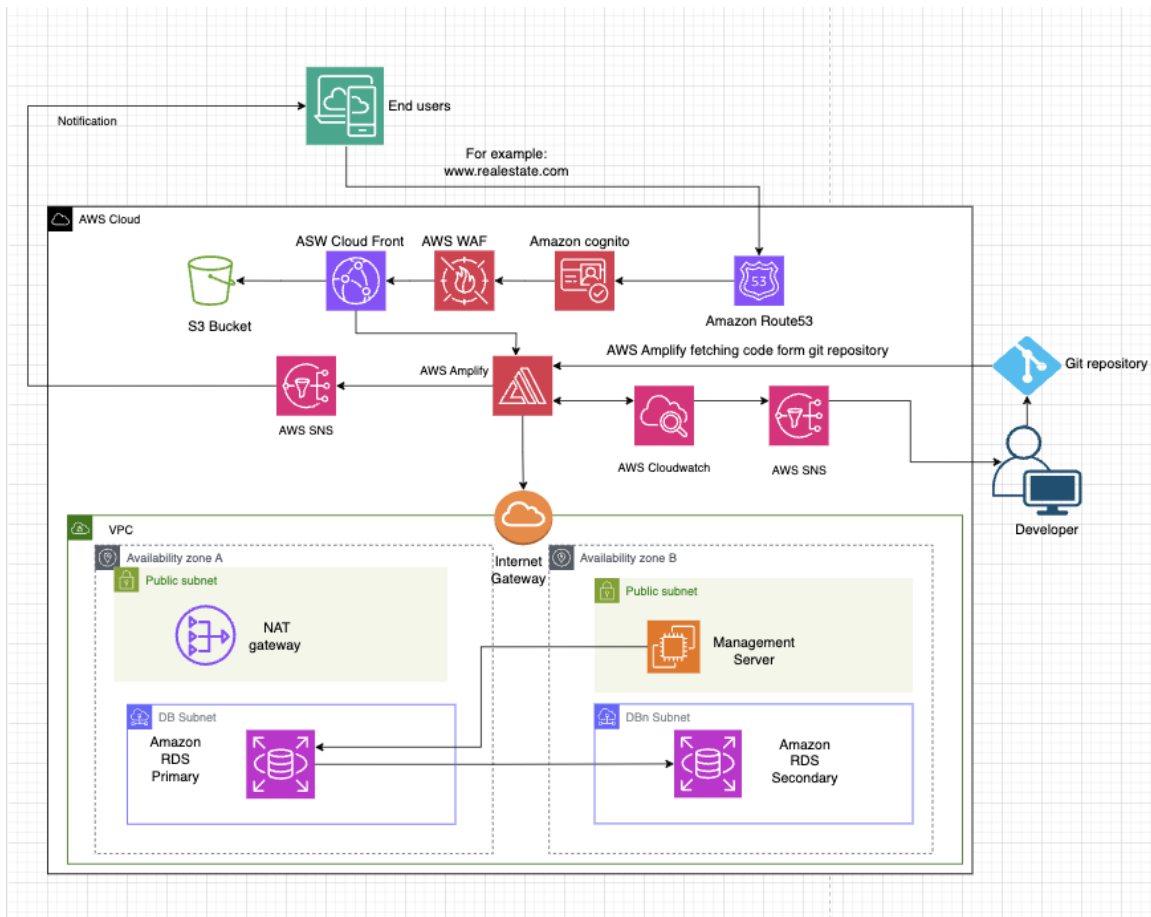


Fig. System architecture design using AWS compute services

The above presented diagram depicts the system architecture which will be hosting the Real Estate application using AWS cloud computer services. This architecture will be highly scalable,

reliable, and cost-effective for small to medium size business, supports handling of traffic during peak hours without compromising performance or user experience. Moreover, other AWS services can be deployed to support the business growth with less administrative work. Below is a brief description of each AWS services with their roles, responsibilities and justification that will be used to host the application:

1. Virtual Private Cloud (VPC)

- a. **Role:** VPC is a logically isolated section of the AWS cloud where several AWS resources can be launched in a virtual network.
- b. **Responsibilities:**
 - Controls network traffic using subnets, security groups, and Network Access Control Lists (NACLs).
 - Enables secure communication between instances within the VPC and the internet, or between instances and other AWS services.
- c. **Justification:** Ensures that the real estate website operates in a secure, isolated environment with full control over networking and security.

2. Availability Zones

- a. **Role:** Physically separate datacenters within a region that provide redundancy and high availability.
- b. **Responsibilities:**
 - Distributes resources across multiple locations to minimize downtime and data loss during failures.
 - Ensures that traffic is spread across multiple instances in different Availability Zones for fault tolerance.
- c. **Justification:** Provides high availability and fault tolerance for the website, ensuring it remains operational even if one AZ experiences an issue.

3. Private Subnets

- a. **Role:** Subnets that host resources not directly exposed to the internet such as database
- b. **Responsibilities:**
 - Ensures that sensitive resources (e.g., databases) are not accessible from the internet, only from within the VPC or via controlled pathways (like the NAT Gateway).
- c. **Justification:** Enhances security by isolating backend components from direct public access, reducing attack surface and protecting sensitive data.

4. Public Subnets

- a. **Role:** Subnets that host resources requiring direct internet access, such as the Application Load Balancer, NAT Gateway and test servers.
- b. **Responsibilities:**

- Hosts components that need to be accessible from the internet, while forwarding requests to backend services hosted in private subnets.
- c. **Justification:** Enables interaction between public-facing resources (like the ALB) and backend services, providing a secure and efficient network design.

5. Amazon Route 53

- a. **Role:** Route 53 is a scalable Domain Name System (DNS) for routing end-users' requests to the web servers.
- b. **Responsibilities:**
 - Routes traffic that is sent from the user to endpoints like EC2, S3, or AWS WAF
 - Provides DNS health checks to ensure that traffic is directed only to healthy endpoints.
 - Manages domains (can register, transfer, and configure DNS settings).
- c. **Justification:** Ensures reliable, low-latency DNS routing to direct users to your application's web servers, contributing to the website's availability

6. AWS WAF (Web Application Firewall)

- a. **Role:** AWS WAF helps to protect the web application from common vulnerabilities like SQL injection, cross-site scripting (XSS), and Distributed Denial of Service (DDoS) attack
- b. **Responsibilities:**
 - Monitors HTTP/HTTPS requests and blocks malicious traffic.
 - Provides granular control to the traffic.
 - Customizable rule sets to filter traffic based on IP address, port number and limit the rate of traffic to avoid DDoS attack.
- c. **Justification:** Enhances the security of the real estate website by protecting it from common to large known attacks like DoS, DDoS, improving security compliance.

7. Amazon Cognito

- a. **Role:** User authentication and identity management service.
- b. **Responsibilities:**
 - Manages user registration, sign-in, and access control, supporting multi-factor authentication (MFA) and integration with social identity providers (like Google, Facebook).
 - Supports millions of users with high performance.
- c. **Justification:** Helps scale the user base securely while providing a seamless login experience for both web and mobile users, ensuring that only authorized users can access secure features of the real estate platform.

8. Amazon CloudFront

- a. **Role:** It is a high-speed and low latency content delivery service.

b. Responsibilities:

- The main function of this service is to cache frequently accessed content at edge location for low-latency delivery.
 - Distributes both static and dynamic content while protecting it with integrated security features like SSL and AWS Shield
- c. **Justification:** Reduces load on backend servers by serving cached content from edge locations, improving the performance and responsiveness of the website, especially for users in geographically distant regions.

9. Amazon S3 (Simple storage Service)

- a. **Role:** A highly scalable object storage service that stores and retrieves any amount of data from anywhere on the web.
- b. **Responsibilities:**
- Stores static assets such as property images, videos, and backups.
 - Can integrate with CloudFront to deliver media content faster.
 - Provides durability, with automatic replication of objects across multiple Availability Zones.
- c. **Justification:** Cost-effective and durable solution for storing large media files (property images), which are crucial to real estate websites.

10. Amazon Amplify

- a. **Role:** Fully managed service for developing and deploying full-stack web and mobile apps.
- b. **Responsibilities:**
- Integrates front-end and back-end components by fetching code from the Git repository for managing continuous deployment
 - Enables easy scaling for both web and mobile applications.
- c. **Justification:** Simplifies app deployment and scaling without requiring infrastructure management, ensuring that the platform remains flexible and can adapt to changing business demands as it grows.

11. Amazon Cloud watch

- a. **Role:** Monitoring and logging service.
- b. **Responsibilities:**
- Provides real-time monitoring of AWS resources and applications, generating alerts based on pre-defined metrics (e.g., CPU utilization, request rates). It also stores log data for troubleshooting.
- c. **Justification:** Enables proactive monitoring and alerting, ensuring that potential issues are detected and addressed before they impact end-user experience, which is crucial for maintaining availability.

12. Amazon SNS (Simple Notification service)

- a. **Role:** Pub/Sub messaging service.
- b. **Responsibilities:**
 - Sends notifications to users or system administrators when specific events occur, such as new property listings, completed transactions, or system failures.
- c. **Justification:** Provides real-time notifications and alerts for critical events, enhancing both user engagement (through updates) and system reliability (through alerts).

13. Git Repository

- a. **Role:** Source code management system.
- b. **Responsibilities:**
 - Hosts the source code of the application
 - Enables developers to collaborate and push updates to AWS Amplify for continuous integration and deployment (CI/CD).
- c. **Justification:** Provides a centralized location for code management, ensuring that updates can be tested and deployed seamlessly via AWS Amplify, improving agility in development and deployment.

14. Internet Gateway

- a. **Role:** A horizontally scaled, redundant, and highly available gateway that allows communication between instances inside VPC and the internet.
- b. **Responsibilities:**
 - Enables outbound internet access for resources in the VPC.
 - Facilitates inbound access from external users to the public-facing services (like the web servers).
- c. **Justification:** Required to enable web servers hosted in the VPC to interact with external users and services over the internet.

15. NAT Gateway

- a. **Role:** A service that enables instances in a private subnet to connect outside of the VPC but preventing access to the private subnet from outside.
- b. **Responsibilities:**
 - Translates private IP addresses into public IP addresses for outbound requests.
 - Allows private ec2 instances to access the internet for software updates or API calls.
- c. **Justification:** Protects private instances from direct internet access while allowing them to make outbound connections as needed (e.g., updates, data fetching).

16. EC2 Instances (Management Servers)

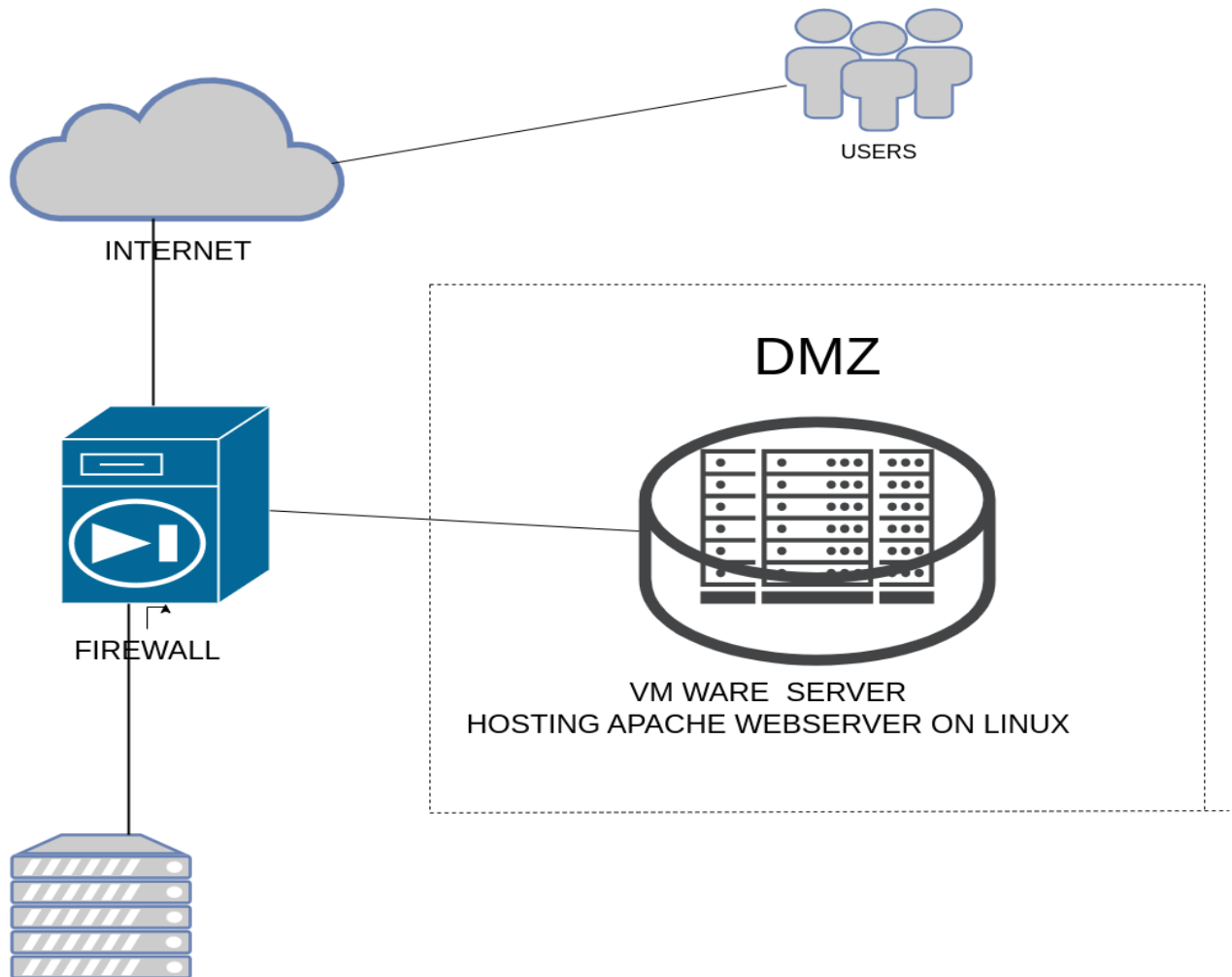
- a. **Role:** For this project the role of the ec2 instance is to manage the database instance which will be only accessible by Management servers for setup and maintenance.

- b. **Responsibilities:**
 - Out-of-band and in-of-band management
- c. **Justification:** Running ec2 instance free tier type is cost-effective and would give enough power for management to the database server.

17. Amazon RDS (Relational Database Service) - Primary and Secondary

- a. **Role:** A managed relational database service with automatic backups, patching, and replication for high availability.
- b. **Responsibilities:**
 - Stores structured data such as user profiles, property listings, and transaction records.
 - Primary database handles read/write operations, while the secondary (replica) provides redundancy and failover capabilities.
- c. **Justification:** Provides a reliable, scalable, and highly available database to ensure consistent performance and data integrity, even during traffic spikes or hardware failures.

2.2 ALTERNATIVE ARCHITECTURE



DATABASE SERVER

Alternative System Architecture: On-Premises Hosting

This section explores the alternative approach of hosting the Real Estate platform using an on-premises infrastructure instead of AWS cloud services. In this model, the infrastructure would be owned and managed internally by the business, including web servers, database servers, and networking equipment.

On-Premises Architecture Overview

The on-premises system architecture involves using physical or virtual servers located within the company's data center. A high-level on-premises architecture design for this Real Estate platform would include the following components:

1. Web Server (Linux with Apache)

a. Role: Hosts the Real Estate website, serving static and dynamic content to end-users.

b. Responsibilities

- Manages HTTP/HTTPS requests and delivers content such as HTML, CSS, and JavaScript.
- Handles requests for backend services like property listings, user profiles, and images.
- **Justification:** Using a Linux-based server with Apache allows for an open-source, cost-effective solution. Apache is a well-established web server that supports large-scale deployments. It's reliable, customizable, and compatible with a wide variety of applications. Apache is a powerful and widely used open-source web server. It can handle large traffic volumes, supports various security modules, and provides SSL support, which is essential for handling sensitive data like user details and transactions on the real estate platform.

2. Database Server (MySQL/PostgreSQL)

a. Role: Manages and stores structured data for the Real Estate platform, such as property listings, user information, transactions, and other records.

b. Responsibilities

- Supports read and write operations from the web server.
- Ensures data integrity and security.
- Performs regular backups for data protection.

c. Justification: A Linux-based server running MySQL or PostgreSQL offers an open-source, highly scalable, and reliable relational database system. Both MySQL and PostgreSQL are excellent choices for handling large datasets, ensuring robust performance and high availability.

3. Network Equipment

a. Role: Ensures seamless communication between the servers, users, and the internet.

b. Responsibilities:

- Provides LAN connectivity between internal resources (e.g., web server, database server).
- Configures switches and routers to manage traffic between the internal network and the external internet connection.
- Secures the network using firewalls, VPNs, and intrusion detection/prevention systems (IDS/IPS).

c. Justification: On-premise hosting requires dedicated network equipment to ensure low latency and reliable connectivity. The use of managed switches, routers, and firewalls would help manage traffic and secure communication.

4. Virtualization Platforms: VMware & Hyper-V

VMware: It can be used to virtualize the physical infrastructure, allowing multiple virtual machines (VMs) to run on a single physical server. Virtualization provides resource optimization, scalability, and easier disaster recovery mechanisms. VMware offers an enterprise-grade virtualization solution. It provides a comprehensive platform for creating and managing virtual environments, allowing businesses to host multiple virtual servers (such as the web and database servers) on fewer physical machines. VMware also supports high availability and fault tolerance.

Usage: Virtualizing the web and database servers would reduce hardware costs and increase flexibility. VMware provides powerful management tools for resource allocation, backups, and failover support.

5. Firewalls & Security (Network Perimeter Security)

a. Role: Protects the on-premise infrastructure from external threats.

b. Responsibilities:

- Configures firewall rules to allow only legitimate traffic.
- Monitors incoming and outgoing traffic for any suspicious activities.

c. Justification: Firewalls, such as pfSense or a proprietary solution, are essential for an on-premises setup to secure network access to the web and database servers. Additionally, an intrusion detection system (IDS) can be deployed to further secure the infrastructure.

6. Backup and Disaster Recovery

a. Role: Ensures that data is regularly backed up and can be restored in case of hardware failure, data corruption, or disaster.

b. Responsibilities

- Implement automated daily backups of the database and web content.
- Set up off-site backups or redundant storage for additional security.

c. Justification: Backups are critical in ensuring data availability in case of server failure or data breaches. Using software like resync or dedicated hardware appliances would ensure that the data remains secure and recoverable.

Advantages of On-Premises Hosting

1. **Full Control:** Complete control over the hardware and software stack. You can customize the infrastructure and security measures according to specific needs.
2. **Data Privacy** Data is stored on your infrastructure, providing better control over data privacy and compliance with regulations (e.g., GDPR).
3. **Cost Management*** Although upfront costs are higher, long-term costs can be managed, particularly when resources are stable and predictable.

Challenges and Drawbacks of On-Premise Hosting

1. **High Initial Cost:** Requires a significant investment in hardware, networking equipment, and data center facilities.
2. **Scalability Issues:** Scaling resources (e.g., adding more servers to handle traffic spikes) may require additional hardware purchases, which takes time and planning.
3. **Maintenance** Requires in-house IT staff for maintenance, updates, backups, and security management.
4. **Disaster Recovery** You must implement your own backup and disaster recovery systems, which may add to the complexity.

Rationale for Choosing AWS Over On-Premise Hosting

While on-premise hosting gives full control and data privacy, AWS cloud services offer several advantages that make them a superior option for this Real Estate platform:

1. **Scalability:** AWS provides instant scalability to handle traffic spikes without the need for additional hardware purchases.
2. **Cost Efficiency:** With AWS, you only pay for what you use. This is particularly useful for small to medium-sized businesses with variable traffic.
3. **High Availability and Redundancy** AWS services like EC2 and RDS are distributed across multiple Availability Zones, providing built-in redundancy and high availability.
4. **Managed Services** AWS offers fully managed services like RDS for databases, which reduce the operational burden on in-house teams. This is particularly valuable for small teams with limited IT resources.

Although an on-premises solution with Linux web servers and Apache provides full control and data privacy, the AWS architecture provides greater scalability, flexibility, and cost-effectiveness, especially for handling unpredictable traffic and reducing operational complexity. Virtualization solutions like VMware or Hyper-V can be useful in the on-premises context, but for this Real


Estate platform, the AWS approach is more aligned with business growth, agility, and reduced maintenance overhead.

2.3 DESIGN PROTOTYPE

Link to GitHub repository: https://github.com/ruff27/fyp_proto

A front-end design has been developed to display how the application would function once implemented. Certain functionalities of the current website will not work as the development has not yet reached the implementation stage. This is just for an overview.

Home page -



Find Your Dream Home


Select Location

Property Type

Price Range (BDT)

Search

Featured Properties




1 Bed Apartment

Location: Dhaka

Price: BDT 8,00,000

View Details




2 Bed House

Location: Chattogram

Price: BDT 12,00,000

View Details



Prime Land

Location: Sylhet

Price: BDT 15,00,000

View Details

Log in and signup page -

UNIT CODE COS40005 EAT40005
PORTFOLIO TASK 3
PAGE 15

Login

Phone or Email:

Password:

Login

[Forgot Password?](#)

Don't have an account? [Sign up here](#)

Signup

Name:

Email:

Phone:

Password:

Confirm Password:

Sign Up

Already have an account? [Login here](#)

Rent/buy page -

Filter Properties

Location

Select Location

Property Type


Select Property Type

Price Range (BDT)

Select Price Range

Search

Available Properties




1 Bed Apartment

Location: Dhaka

Price: BDT 8,00,000

View Details




2 Bed House

Location: Chattogram

Price: BDT 12,00,000

View Details

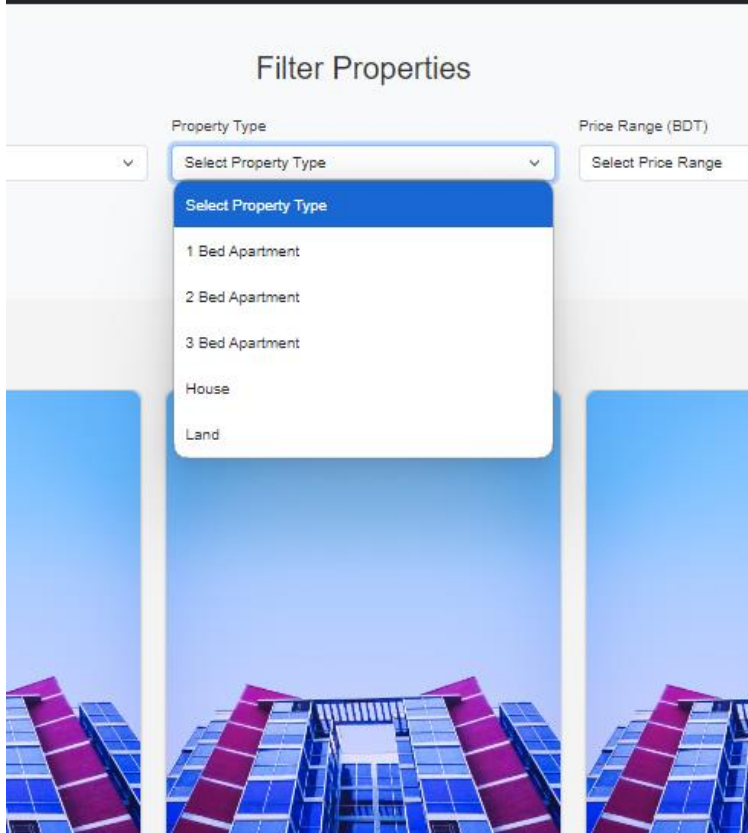
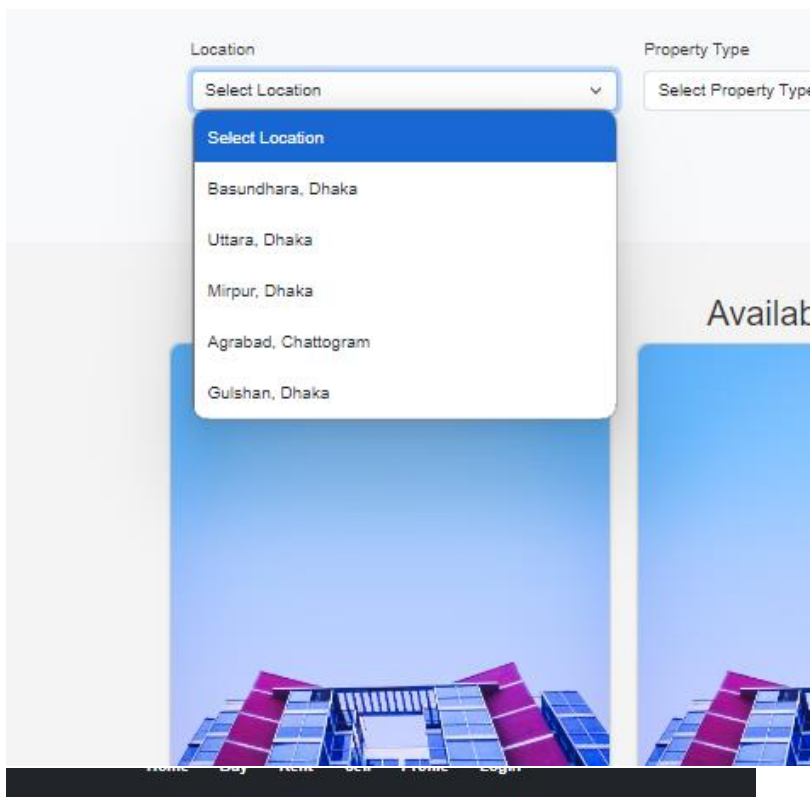


Prime Land

Location: Sylhet

Price: BDT 15,00,000

View Details



Submit listing

The screenshot shows a web application with a dark navigation bar at the top containing links: Home, Buy, Rent, Sell, Profile, and Login. Below this is a light gray box titled "Sell Your Property". The form contains the following fields:

- Property Title:** A text input field with the placeholder "Enter property title".
- Location:** A dropdown menu with the placeholder "Select Location".
- Property Type:** A dropdown menu with the placeholder "Select Property Type".
- Price (BDT):** A text input field with the placeholder "Enter price".
- Property Description:** A large text area with the placeholder "Enter property description".
- Property Image:** A file upload section with a "Choose File" button and the text "No file chosen".

At the bottom of the form is a dark blue button labeled "Submit Listing".

Chat interface -

The screenshot shows a chat window titled "Chat with Abdul" with a close button (X) in the top right corner. The chat history shows:

- You:** Hi, I would like to inspect this property
- Abdul:** Thank you for your message. I'll respond shortly.

At the bottom, there is a text input field with the placeholder "Type your message" and a blue "Send" button.

3. RESEARCH AND INVESTIGATIONS

The architectural design and technology stack for this real estate platform were meticulously selected based on rigorous performance benchmarks and scalability requirements. The core

technologies—Amazon Web Services (AWS) for cloud infrastructure, ReactJS for frontend development, MySQL for data persistence, and Node.js for backend services—were chosen after comparative analysis against alternatives. AWS provides a robust, scalable cloud environment, ReactJS offers a component-based architecture for efficient UI development, MySQL ensures ACID-compliant data storage, and Node.js enables high-performance, event-driven server-side operations.

During the requirements analysis and system design phases, extensive research was conducted into the technical landscape of the Bangladeshi real estate market. This investigation revealed critical gaps in existing platforms like Tolet and BD Housing, particularly in areas of real-time data synchronization, geospatial integration, and user authentication protocols. To address these deficiencies, the research focused on implementing WebSocket technology for real-time property updates, integrating the Google Maps API for interactive geospatial features, and designing a multi-factor authentication system to enhance platform security and user trust.

Architectural considerations led to the adoption of a monolithic design, leveraging AWS services such as Amplify for streamlined frontend deployment, Amazon RDS for managed MySQL instances, and S3 for scalable object storage. This architecture facilitates easier initial deployment and management while allowing for future microservices migration if required. The research also explored implementing a Cache-Aside pattern using Amazon ElastiCache to reduce database load and improve response times for frequently accessed data.

Security research was paramount, focusing on compliance with the Digital Security Act of Bangladesh and international data protection standards. The implementation plan includes AES-256 encryption for data at rest and TLS 1.3 protocols for data in transit. AWS Identity and Access Management (IAM) will be utilized to implement the principle of least privilege, while Security Groups will provide network-level isolation. Additionally, the research proposed implementing JSON Web Tokens (JWTs) for stateless authentication, supplemented by OAuth 2.0 for third-party integrations.

User experience (UX) and user interface (UI) research emphasized the importance of responsive design and progressive enhancement techniques to ensure optimal performance across various devices, particularly mobile platforms which dominate the local market. The frontend architecture will utilize React's Context API for state management, with Redux considered for more complex state scenarios. Internationalization will be implemented using the i18next library, allowing seamless switching between English and Bangla interfaces. The research also explored implementing a custom Bangla input method editor (IME) to facilitate easier text entry on various devices.

Geospatial functionality emerged as a critical feature, with research focusing on efficient integration of the Google Maps JavaScript API. Plans include implementing custom overlays using SVG and the Maps API's drawing tools to represent property boundaries. On the backend, the research proposed using MySQL's spatial extensions for geospatial indexing and querying, enabling efficient proximity-based searches.

Performance optimization strategies were extensively researched, including the implementation of a content delivery network (CDN) using Amazon CloudFront to reduce latency. The research also explored implementing server-side rendering (SSR) with Next.js to improve initial page load times and search engine optimization (SEO). Lazy loading techniques for images and below-the-fold content were identified as crucial for improving perceived performance, especially on slower mobile networks prevalent in the target market.

This comprehensive technical research lays the groundwork for developing a robust, scalable, and secure real estate platform tailored to the unique requirements of the Bangladeshi market. The proposed architecture and technology choices aim to address current market gaps while providing a flexible foundation for future enhancements and scalability.

3.1 RESEARCH INTO THE APPLICATION DOMAIN

1) Overview of Existing Real Estate Apps and Websites in Bangladesh

Several platforms in Bangladesh cater to real estate needs, offering property listings and rentals. The most relevant platforms include:

Property
Tolet
BDHousing

Each of these platforms provides users with a variety of property options, but they vary significantly in user experience, feature set, and performance. By understanding their limitations, we can design our app to offer a better, more focused experience, particularly for students.

2) Analysis of Existing Platforms

2.1) BProperty

Strengths:

Established platform with a large database of properties across major cities.

Provides detailed property listings with images, descriptions, and prices.

Allows users to schedule site visits directly through the platform.

Weaknesses:

Does not allow users to sell properties, limiting functionality.

The user interface is cluttered and not particularly student-friendly, which makes it harder for specific demographics to find affordable accommodations.

May experience performance issues during high-traffic periods, as there are no clear backend optimizations.

2.2) Tolet

Strengths:

Focuses on rentals, especially in urban areas, which is highly relevant to students seeking accommodation.

Simple interface for quickly listing rental properties.

Weaknesses:

Limited property listings in high-demand areas like Dhaka and Chittagong.

Lacks features like proximity-based search or advanced filtering for different user needs.

No built-in secure communication channels.

Performance under heavy load may suffer as there are no evident optimizations for handling high traffic.

2.3) BDHousing

Strengths:

Provides listings for residential, commercial, and rental properties.

Features detailed descriptions of properties.

Weaknesses:

Limited number of properties, particularly in non-urban areas.

The platform's design is outdated, making it less appealing for mobile-first users, especially students.

No evident infrastructure to manage traffic surges, which could lead to slowdowns during peak periods.

3) Gaps and Opportunities**3.1) Several key gaps can be addressed:**

- **Student-Centric Listings:** None of these platforms have a dedicated section for student accommodation, which are often mixed with more expensive listings. Our platform will isolate student listings, making it easy for them to find affordable housing.
- **Performance Issues Under High Traffic:** Given Bangladesh's large real estate market and seasonal traffic spikes (especially during academic seasons), these platforms may experience performance issues without proper traffic management tools like load balancing and congestion control.
- **Proximity-Based Search:** Current platforms do not allow users to search based on proximity to key locations like universities and schools. Integrating Google Maps for proximity-based search is crucial for students seeking accommodations near educational institutions.
- **Secure Communication Channels:** Most platforms lack encrypted communication, which is critical for secure negotiations between buyers, sellers, and renters.

3.2) Proposed Solutions with AWS Services

- Our platform will address these gaps by providing student-centric listings, proximity-based search, and optimizations for high traffic. We will use AWS services to ensure the platform is scalable, secure, and responsive. Some AWS services may incur costs beyond the Free Tier, so those will be marked as out of scope for now but discussed for future use.

- **Student-Centric Listings with Affordability Filters:** The platform will have a dedicated section for student-targeted listings, ensuring they do not get lost among high-end properties. Affordability filters and proximity-based searches will help students quickly find accommodations near their institutions.
- **Proximity-Based Search via Google Maps:** Google Maps will be integrated to allow users to search based on proximity to schools, universities, hospitals, and transport hubs. This will particularly benefit students and make the search process smoother.

Optimized Performance with AWS Load Balancing:

Using AWS Elastic Load Balancer (ELB), we will distribute incoming traffic across multiple EC2 instances to prevent server overload. ELB, while not part of the Free Tier, offers a scalable solution for managing traffic surges. For now, it is out of scope but remains a viable solution as the platform grows.

Auto Scaling, which is available in the Free Tier, will dynamically add or remove EC2 instances based on traffic. This ensures we can handle surges in demand without manual intervention.

AWS-Managed Congestion Control:

AWS Global Accelerator, while out of scope, is an excellent tool for improving response times by routing user traffic to the nearest available server region. This service helps minimize latency across a large user base.

For real-time data transmission, AWS supports TCP congestion control algorithms like Cubic or Reno to manage how data packets are delivered during traffic spikes. Using these with AWS CloudFront can improve the speed of data transfer during peak usage, which is especially important for our student-heavy user base.

Rate Limiting with AWS API Gateway:

AWS API Gateway, which is part of the Free Tier, will help us manage and throttle API requests to ensure no single user overloads the system. This ensures the app remains responsive, even under heavy load.

Secure, Encrypted Communication:

We will use AWS Key Management Service (KMS), available under the Free Tier, for managing encryption keys securely. Alongside CryptoJS, we will implement real-time encrypted chat, ensuring secure communication between students and landlords, as well as other users.

AWS Secrets Manager will be considered for production use but is out of scope due to potential costs.

Mobile-First Design with AWS Amplify:

AWS Amplify, which is part of the Free Tier, will be used to build and deploy a mobile-first app. This ensures students, who primarily use mobile devices, have a seamless experience. Amplify integrates well with other AWS services like Cognito for authentication, providing a smooth development workflow.

4) Conclusion

Our platform will not only cater to the specific needs of students but will also address broader real estate needs, providing a secure, scalable, and high-performing platform. By using AWS Elastic Load Balancer (out of scope for now) and Auto Scaling (within scope), we can ensure that the platform performs well under heavy traffic, particularly during student-heavy seasons.

Additionally, AWS Global Accelerator (out of scope) and congestion control algorithms will manage network traffic, ensuring smooth data delivery. Rate limiting with API Gateway will prevent the system from being overwhelmed, while CryptoJS and KMS will ensure that user communications are secure and private.

All of these services are built with AWS, leveraging the Free Tier for development and testing, while more advanced services (though out of scope for now) are available for future scalability. This will allow us to offer a platform that not only meets the needs of students but also outperforms existing alternatives in terms of performance and user experience.

3.2 RESEARCH INTO SYSTEM DESIGN

While designing a system architecture for a real estate website that will be in practice in Bangladesh a developing country several key factors are considered such as availability,

scalability, security, performance and cost-effectiveness. The traditional way of hosting web applications requires high upfront and maintenance costs, so we orchestrated system design in AWS cloud platform which is pay-as-you-go model and even some services are free to use. Below is a deeper exploration of how AWS services can leverage to build an optimized, scalable and secure system to host a real estate website targeted for Bangladesh.

Key considerations for the AWS architecture:

a. Availability:

AWS cloud services are made to be highly available in different geographical regions and compatible with network needs. Moreover, cloud service is available from any location and at any time can significantly enhance the operational availability of organizations and industries in Bangladesh (Islam et al., 2017). As of 2024 AWS has 108 availability zone (AWS 2024) including 2 zones in Indian which is near to Bangladesh.

b. Scalability & Traffic Handling:

The designed AWS System is highly scalable, the AWS amplify can cope with the growth of the traffic and automate scale to provide service without interruption user experience. Furthermore, AWS Amplify is a fully managed service that provides backend and frontend development capabilities without the need to manage servers like EC2. It has built in typescript, sever framework for developing application which is best suited for the developer.

c. Cost Efficiency:

Developing countries often demand cost-effective solutions to be competitive. Building up physical servers requires huge upfront, and maintenance cost, using AWS cloud infrastructure the maintenance responsibility does not fall into the us which saves labour costs and huge amount of time. AWS offers a pay-as-you-go pricing model, enabling businesses to only pay for the resources that has been used, which is ideal for startup like ours or small to medium-sized enterprise (SMEs) with tight budgets.

d. Latency and User Experience:

Considering Bangladesh's network infrastructure, minimizing latency is critical. AWS's edge services like CloudFront will help reduce latency by delivering cached content from the nearest edge locations, improving the user experience across varying internet speeds. Since, the AWS has high geographical area coverage, users will get the service with low latency through good internet connectivity.

e. Security and Compliance:

Given the increasing cybersecurity threats, protecting sensitive user data (e.g., personal information and transaction records) is critical. The Amazon Cognito enhance the authentication process of the user sing in and sing up. Moreover, utilizing AWS services

like VPC, AWS WAF, and security groups will ensure that security is handled effectively, reducing the risk of breaches.

f. Adaptability

As on my research there are many companies including banks who have already migrated to cloud environments. “Companies like ACI, Robi, Eastern Bank etc. are few of the so numerous mammoth companies who have considered having cloud administrations in their company” (Yusuf, Hasan, Adan and Mohamoud 2022). Since, the AWS cloud infrastructure has flexibility the project would benefit for the deployment in Bangladesh.

3.3 RESEARCH INTO TECHNICAL PLATFORM

The technical platform for the real estate application targeting the Bangladeshi market is designed to deliver a secure, scalable, and localized solution that supports real-time updates, trust-building, and affordability. The selected tools and services focus on addressing the specific needs of the platform’s users: real estate professionals, buyers, sellers, and students looking for properties.

1. Frontend Technologies

The frontend of the platform needs to provide a smooth user experience, support localization, and handle real-time interactions effectively.

- 1.1 Web Frontend: React.js
Reasoning: React.js is chosen for its ability to create fast, responsive, and dynamic web interfaces. It easily integrates with third-party services like Google Maps API to offer an interactive property search. Additionally, localization (English and Bengali) is supported using libraries like react-i18next, making the platform accessible to a wide audience.
Key Feature: Fast real-time updates for property listings and support for multi-language functionality.
- 1.2 Mobile Frontend: React Native
Reasoning: React Native allows the platform to be deployed on both Android and iOS, which is critical for Bangladesh, where mobile devices are the primary access point to the internet. A mobile-first design ensures that users can search, view, and interact with listings from anywhere.
Key Feature: Cross-platform development ensures efficient and fast mobile app delivery, targeting Bangladesh’s mobile-centric users.

2. Backend Technologies

The backend handles user interactions, property listings, and verification processes in real-time while ensuring scalability and security.

- 2.1 Backend Framework: Node.js with Express.js

Reasoning: Node.js is ideal for handling real-time data and high traffic loads with its non-blocking architecture. It supports real-time chat and property updates through Web Sockets, ensuring seamless communication between users and agents. Express.js provides a simple framework for managing API requests, making it easier to scale as user demand grows.

Key Feature: Real-time communication for users and agents, ensuring quick interaction and property status updates.

3. Cloud Infrastructure: Amazon Web Services (AWS)

AWS offers a reliable and scalable infrastructure to support the platform, particularly important for handling high traffic and ensuring security.

- 3.1 AWS Amplify

Reasoning: AWS Amplify simplifies integration with backend services and supports fast, scalable deployment. It also provides real-time capabilities, ensuring users get instant notifications about property updates or responses from agents.

Key Feature: Seamless integration for real-time property updates and notifications.

- 3.2 Amazon RDS (Relational Database Service)

Reasoning: RDS provides a managed relational database for storing structured data like property listings, user profiles, and transactions. It supports automatic backups and can scale as the platform grows.

Key Feature: Reliable and scalable database management with automated backups.

- 3.3 Amazon S3 (Simple Storage Service)

Reasoning: Amazon S3 stores property images and videos securely and at scale, ensuring that users can easily view rich media content without impacting performance.

Key Feature: Scalable and cost-effective media storage for property listings.

- 3.4 Amazon CloudFront (CDN)

Reasoning: CloudFront reduces the loading time for users by caching property images and videos across multiple regions, improving performance for users across Bangladesh.

Key Feature: Faster content delivery through global edge locations, improving user experience in remote areas.

4. Security Features

Given the need to build trust in the platform, security is a priority.

- 4.1 AWS WAF (Web Application Firewall)
Reasoning: AWS WAF protects the platform from common threats like SQL injection and DDoS attacks, which are crucial to maintaining trust in real estate transactions.
Key Feature: Ensures protection against common vulnerabilities, keeping user data secure.

5. Third-Party API Integrations

The platform integrates third-party APIs to provide location-based search and real-time communication.

- 5.1 Google Maps API
Reasoning: Google Maps enables location-based property searches, allowing users to find properties near landmarks such as schools and universities, particularly useful for students.
Key Feature: Interactive map-based search, helping users visualize properties in relation to their surroundings.
- 5.2 Firebase Cloud Messaging (FCM)
Reasoning: FCM supports real-time push notifications, alerting users to new listings or updates to favorite properties.
Key Feature: Real-time notifications for property updates and agent responses, enhancing user engagement.

3.4 OTHER RESEARCH

User Survey on Key Features for Renters and Owners

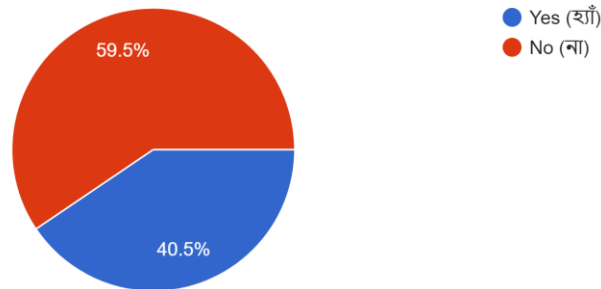
To gather deeper insights for the development of the real estate platform, we conducted a comprehensive survey targeted at both property renters and owners in Bangladesh. This survey was crucial to understanding the specific needs and preferences of users, particularly in areas such as property listing features, user interaction preferences, and safety concerns.

From the data collected, several key trends emerged:

- **Anonymous Posting Concerns:** 59.5% of renters expressed concerns about trusting anonymous property postings, highlighting the importance of user verification. The need for transparency was echoed throughout the survey.

As a renter, would you trust anonymous postings? একজন ভাড়াটিয়া হিসেবে, আপনি কি গোপনীয় পোস্টিং বিশ্বাস করবেন?

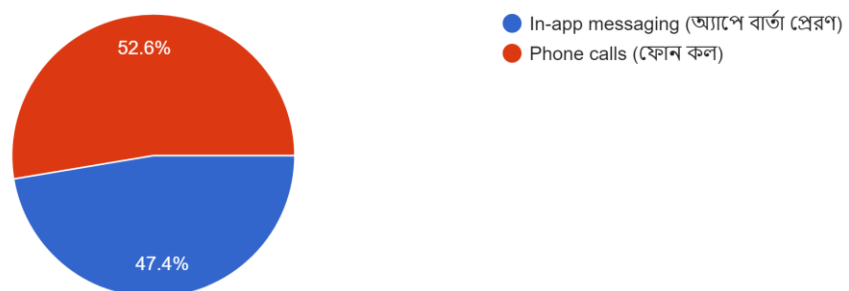
37 responses



- **Room Sharing Features:** A significant portion of respondents raised concerns about safety when asked whether they would be comfortable sharing rooms with strangers. These insights suggest that while room sharing could be a useful feature, security and trust-building mechanisms, such as verification, will be vital.
- **Communication Preferences:** 52.6% of renters and 55.2% of owners prefer phone calls for communication, with a significant portion also favoring in-app messaging. This emphasizes the need to build a flexible communication system that can cater to both preferences.

Would you prefer to communicate with property owners via in-app messaging or phone calls? আপনি কি অ্যাপের মাধ্যমে বার্তা প্রেরণের ম...সাথে যোগাযোগ করতে পছন্দ করবেন, নাকি ফোন কলে?

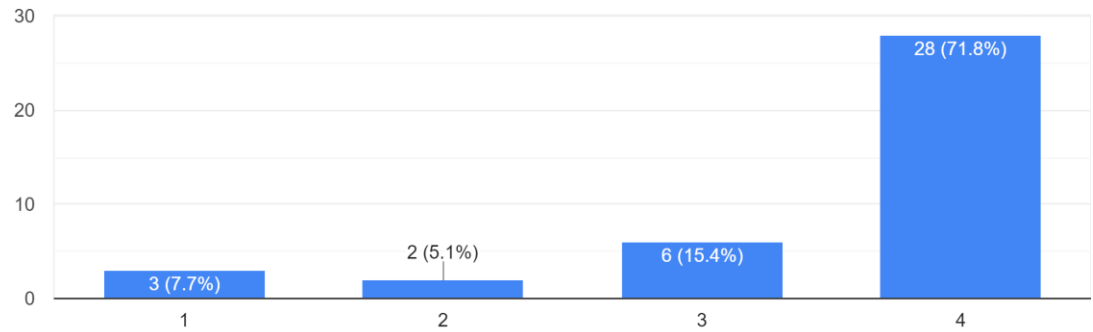
38 responses



- **Property and Tenant Verification:** 71.8% of renters and 82.1% of owners highlighted the importance of verification, making it a critical feature to include in the platform. Ensuring both property and tenant credibility emerged as a top priority for the users.

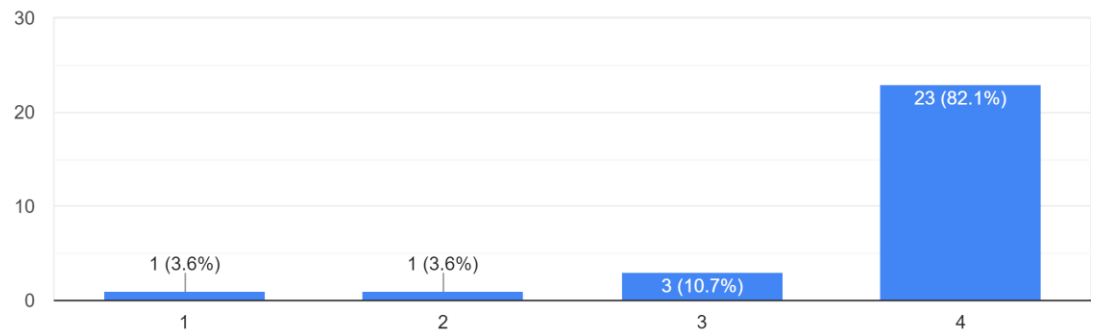
How important is property verification for you in a real estate app? রিয়েল এস্টেট অ্যাপে সম্পত্তির যাচাই আপনার জন্য কতটা গুরুত্বপূর্ণ?

39 responses



How important is tenant verification for you? ভাড়াটিয়ার যাচাই আপনার জন্য কতটা গুরুত্বপূর্ণ?

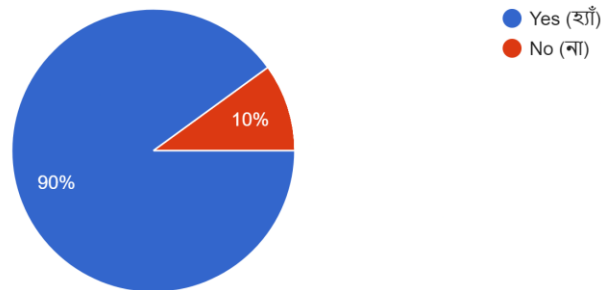
28 responses



- Scheduling Visits: 90% of owners supported the option to schedule property visits via the app, further emphasizing the need for integrating appointment management features.

Would you like an option to schedule property visits through the app? আপনি কি অ্যাপের মাধ্যমে প্রপার্টি ভিজিটের সময়সূচি নির্ধারণ করার বিকল্প চান?

30 responses



These findings will directly influence the design of features like property verification, user reviews, and secure communication channels.

Color Research

The color scheme was a critical aspect of our platform's user experience design, especially considering the prevalence of color vision deficiency (CVD) in Bangladesh. A study conducted in 2013 on medical and dental students at Mymensingh Medical College found that 3.35% of the population, particularly males, suffer from red-green color blindness. Given this data, we sought the expertise of UI/UX professionals from both Bangladesh and Australia, specializing in app building and game development, to guide the color selection process.

The survey results and professional input led to the conclusion that color palettes avoiding red and green dominance would be most suitable for our platform. The final colors were chosen to ensure visual accessibility for a broad audience, focusing on shades that are easily distinguishable even by users with CVD. This approach is in line with international best practices for accessibility, ensuring the platform remains functional and appealing to all users.

Collaboration between experts from both countries, leveraging their experience in local and global markets, allowed us to develop a balanced and inclusive color scheme. The selected palette supports a wide range of user preferences, while ensuring that the platform remains visually accessible for those with color vision deficiencies.

This research also included a comparative study of CVD prevalence in Bangladesh and other countries. The goal was to evaluate whether Bangladesh has a higher or lower CVD rate compared to the global average. The findings indicate that red-green CVD is the most common type, consistent with global trends. As a result, our decision to avoid red-green-heavy designs is further supported by both local and international data, reinforcing the inclusiveness of our design choices.

Color Theme Options:



A



B



C

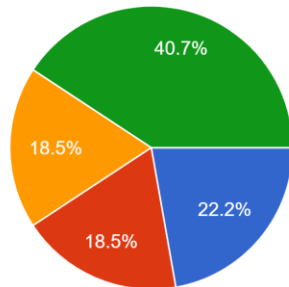


D



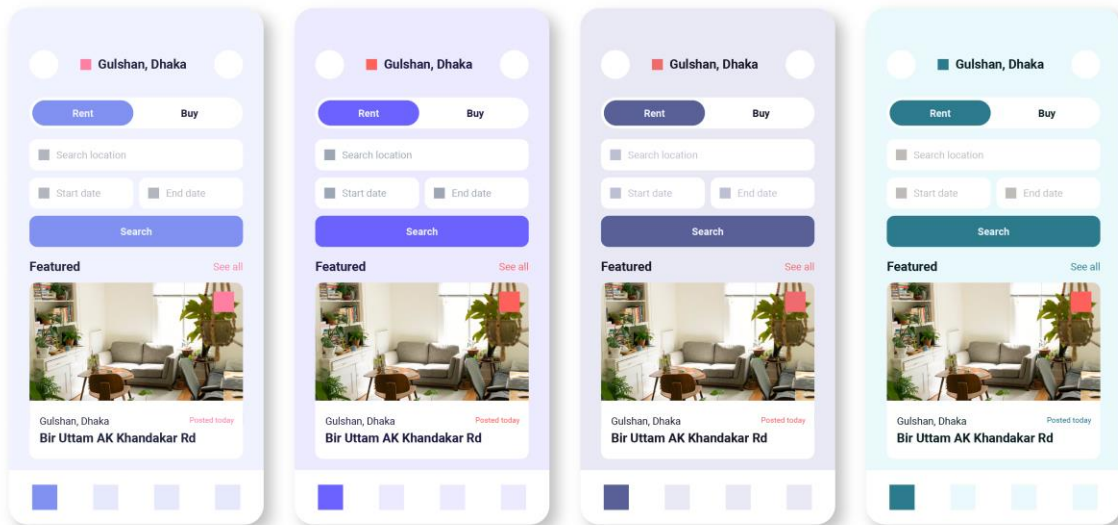
What color theme would appeal to you for the app? অ্যাপের জন্য আপনার কোন রঙের থিম আকর্ষণীয় মনে হয়?

54 responses



- A
- B
- C
- D

UI Color Variations:



4. KOST ANALYSIS

The development of the real estate platform for the Bangladeshi market requires a detailed evaluation of the team's Knowledge, Skills, and Technology (KOST). This analysis ensures that the selected architecture aligns with the team's capabilities while identifying gaps and formulating strategies to overcome them.

Knowledge

The team possesses a strong foundational knowledge of web and mobile application development. This includes understanding the real estate domain through research into platforms like *Tolet* and *BD Housing*. However, the team has limited experience with the specific needs of the Bangladeshi real estate market, such as the cultural nuances in property transactions and the trust challenges posed by unverified listings. To address this gap, the team has consulted with stakeholders familiar with the local market and conducted extensive research into user behavior in Bangladesh (Rahman & Sarker 2023). Additionally, the team plans to gather user feedback through surveys and focus groups to further understand the local market requirements.

Skills

The team has solid skills in backend development, particularly in using Node.js and React.js for handling server-side tasks, and React.js for frontend development. They also have experience in database management using MySQL, making them well-suited to implement the platform's core functionalities like user authentication, property listings, and secure messaging. However, there is a skill gap in advanced cloud architecture, particularly in using AWS services like Lambda for serverless computing and S3 for media storage. To address this, the team will undertake targeted AWS training using online resources and official documentation. Additionally, there is a minor skill gap in integrating third-party APIs such as Google Maps for the interactive property search, which the team plans to fill by utilizing open-source libraries and tutorials (AWS 2024).

Technology

The chosen technology stack includes React.js for frontend development, Node.js and Express.js for the backend, AWS for cloud infrastructure (Amplify, RDS, Lambda, and S3), and Google Maps API for location-based services. This technology stack is aligned with the team's skill set, with React.js and Node.js being familiar tools that the team has used in previous projects. AWS was selected for its scalability and cost-effectiveness, especially under the free tier for early-stage development. The main technology gap lies in managing serverless architecture (AWS Lambda) and efficient media storage (S3). To bridge this gap, the team plans to leverage AWS's extensive documentation and support forums to gain proficiency in these areas.

Gap Plan

While the team has adequate knowledge and skills in most areas, the primary gaps are in AWS cloud services, particularly Lambda and S3, and advanced Google Maps integration. To address this, the team will enroll in AWS training courses and consult the AWS Solutions Architect guides for best practices. Additionally, the team will allocate time for prototyping and testing third-party API integrations, ensuring seamless map-based property search features. With these measures in place, the team is confident in overcoming the skill and technology gaps and delivering a robust, scalable platform.

5. REFERENCE

AWS 2024, Cloud Computing Services, Amazon Web Services, viewed 22 September 2024, <https://aws.amazon.com/>.

AWS, AWS Global Infrastructure, Amazon Web Services, 2024, viewed 23 September 2024, <https://aws.amazon.com/about-aws/global-infrastructure/>.

AWS, AWS WAF, Amazon Web Services, 2024, viewed 22 September 2024, <https://aws.amazon.com/waf/>.

AWS, Amazon CloudFront, Amazon Web Services, 2024, viewed 22 September 2024, <https://aws.amazon.com/cloudfront/>.

AWS, Amazon RDS, Amazon Web Services, 2024, viewed 22 September 2024, https://aws.amazon.com/rds/?gclid=Cj0KCQjwgL-3BhDnARIsAL6KZ6-iLlj_h4WnILk_GJv6nIUJwQNuHXMex4Efdjb9ktqzNmB3zNXpRsaAnt6EALw_wcB&trk=f8c749c2-a797-41e7-9b4d-743b10b206a3&sc_channel=ps&ef_id=Cj0KCQjwgL-3BhDnARIsAL6KZ6-iLlj_h4WnILk_GJv6nIUJwQNuHXMex4Efdjb9ktqzNmB3zNXpRsaAnt6EALw_wcB:G:s&s_kwid=AL!4422!3!549058196525!e!!g!!amazon%20rds!11539887576!114142396522.

AWS, Amazon Route 53, Amazon Web Services, 2024, viewed 22 September 2024, <https://aws.amazon.com/route53/>.

AWS, An AWS Cloud Architecture for Web Hosting, Amazon Web Services, 2024, viewed 22 September 2024, <https://docs.aws.amazon.com/whitepapers/latest/web-application-hosting-best-practices/an-aws-cloud-architecture-for-web-hosting.html>.

AWS, Building a Containerized and Scalable Web Application on AWS, Amazon Web Services, 2024, viewed 22 September 2024, <https://aws.amazon.com/solutions/guidance/building-a-containerized-and-scalable-web-application-on-aws/>.

Bangladesh Journals Online. (2013). Prevalence of colour blindness in young Bangladeshis medical students. Bangladesh Journals Online. Retrieved from <https://www.banglajol.info>

Barua, S., Mridha, A. H. A. M., & Khan, R. H. (2010). Housing real estate sector in Bangladesh: Present status and policy implications. *ASA University Review*, Vol. 4(No. 1), 2010.

Islam, M.A., Kasem, F.B.A., Khan, S., Habib, M.T. and Ahmed, F., 2017. Cloud computing in education: Potentials and challenges for Bangladesh. *International Journal of Computer Science, Engineering and Applications*, 7(5), pp.11-21

Khaled, M. C., Sultana, T., Biswas, S. K., & Karan, R. (2012). Real estate industry in Chittagong (Bangladesh): A survey on customer perception and expectation. *Developing Country Studies*, Vol. 2(No. 2), 38-45. Retrieved from <http://www.iiste.org>

Rahman, A & Sarker, M 2023, Emerging Trends in Real Estate Technology in Bangladesh, *Bangladesh Journal of Economic Studies*, vol. 12, no. 1, pp. 77-89.

Yusuf, A.H., Adan, J., Hasan, F. and Mohamoud, H. (2022). Case study usage and challenges of AWS in Bangladesh and in Somalia. [online] pp.1–25. Available at: https://www.researchgate.net/publication/359352577_Case_study_usage_and_challenges_of_AWS_in_Bangladesh_and_in_Somalia [Accessed 23 Sep. 2024].

- CLIENT SIGN OFF

Name	Position	Signature	Date
Organization			