PORTFOLIO WEBSITE FOR NEONTECH

MD. KHAIRUL BASHAR MAHI

FULL-STACK DEVELOPER

VERSION-5 (Final)

CONTENTS

MD. TARIKUR RAHMAN

Managing Director

UCC

Al Raji Complex,15th Floor.

166-167, Shaheed Syed Nazrul Islam Sarani, Purana Paltan

Dhaka -1000

Dear MD. TARIKUR RAHMAN,

Re: Enclosed Application Agreement for **Online Shopping Website and Web Application System**.

At Neon-Tech, we are aware that creating client-oriented Application takes a mixture of technical excellence and clear communication and our firm hires only the very best to ensure you receive both. We know that every client is unique and we strive to deliver an individual, innovative and affordable proposal every time and to follow it through with an outstanding delivery which is both on time and within budget.

According to your business requirement we design **Online Shopping Website and Web Application System** for you. In this agreement paper we right each and every part and all possible feature which covered our application.

So please read carefully and if it satisfies you then sign and confirm the agreement.

Yours Truly,

Md. Khairul Bashar

Assistant Head

Application Development Brach

Neon-Tech

1. PROJECT OVERVIEW

In 21st century, advantage of the technology and digitalization efforts. The use of technology all of our life has been evolved. To make our business or professional life easy we develop a system. This project consists a website and a web application, Website is fully information based and the web application has two modules with cloud features.

2. OBSTACLES

- Low Budget
- Platform
- Manage Storage File System

3. USER FEATURES IN DETAILS:

Online Shopping Website and Web Application System has two parts, They Are:

- Website
  - Home
  - Events
  - Video & Photo Gallery
    - Admin Panel only
      - Can upload photos and Video
  - Blog
  - Achievements
  - Projects
  - About Us
  - Login (Go for The Web Application)
- Web Application System
  - User
    - Login
      - User Id and Password are provided by admin
  - Admin
    - Create User
    - Create Events See
    - Upload blogs
    - Upload Photos & Videos

4. TECHNOLOGY REQUIRE (SOFTWARE AND HARDWARE)

- Framework: Slim.
- Database: MySQL.
- Design: Standard.
- Coding Architecture: OOP.
- Security: Standard.

4.1 SOFTWARE REQUIREMENTS

Front-end Language:

For front-end development:

- HTML5
- CSS3
- JavaScript (often with frameworks/libraries like React, Angular, or Vue.js)

Back-end Language:

For MVC V5 model, common choices:

- ASP.NET (C#)
- ASP.NET Core

Database:

SQL Server is a popular choice for .NET applications. Other options:

- MySQL
- PostgreSQL
- MongoDB (for NoSQL database)

Security:

Implement security best practices, including:

- HTTPS for secure communication
- Input validation and sanitization
- Authentication and authorization mechanisms
- Regular security audits and updates

Model:

Model-View-Controller (MVC) Version 5 is associated with ASP.NET. Ensure you follow MVC architectural patterns for organizing code.

4.2 HARDWARE REQUIREMENTS

- **Memory Size:** The memory requirements depend on the expected user load and the complexity of your application. Start with a scalable hosting solution that allows adjustments based on usage.
- **Bandwidth:** Bandwidth needs are influenced by the number of users, the content on your website, and the frequency of updates. Choose a hosting provider that offers sufficient bandwidth to handle your expected traffic.

5. APPLICATION

- **Online Shopping Website and Web Application System.**

6. MILESTONES AND REPORTING

Milestone	Tasks	Reporting	Time
Analysis		Submit The Design	3 days
Requirements Collection	Submit To Us All Data		7 days
Development		Review The Work	30 days
Testing			10 days
Deployment	Must Ready the Server	Review Final Work	5 days
Delivery		Live On Server	5 days

7. SDLC

Certainly! Here are some common Software Development Life Cycle (SDLC) models, along with their descriptions, typical use cases, and phases:

7.1 WATERFALL MODEL

- Description: The Waterfall model is a linear and sequential approach to software development, where each phase must be completed before the next one begins.

- Use: It is suitable for projects with well-defined and stable requirements.

- Phases:

 1. Requirements Analysis

 2. System Design

 3. Implementation

 4. Integration and Testing

 5. Deployment

 6. Maintenance

7.2 ITERATIVE MODEL

- Description: The Iterative model involves repeating cycles (iterations) of the development process. Each iteration includes phases from requirements to testing.

- Use: It is useful when requirements are not clear or may change during the project.

- Phases: Similar to the Waterfall model but repeated in cycles.

7.3 SPIRAL MODEL

- Description: The Spiral model combines iterative development with elements of the Waterfall model. It emphasizes risk assessment and mitigation.

- Use: It is suitable for large and complex projects with evolving requirements.

- Phases:

  1. Planning

  2. Risk Analysis

  3. Engineering (Design, Code, Test)

  4. Evaluation (Review and Testing)

  5. Repeat Iterations (If Necessary)

  6. Deployment and Maintenance

7.4 V-MODEL (VALIDATION AND VERIFICATION MODEL)

- Description: The V-Model is an extension of the Waterfall model that emphasizes validation (testing) at each phase.

- Use: It is used when high reliability and thorough testing are essential.

- Phases: The phases are similar to the Waterfall model, but testing is closely aligned with each development phase.

7.5 AGILE MODEL

- Description: Agile is an iterative and flexible approach that focuses on collaboration, customer feedback, and delivering working software in short iterations.

- Use: It is best for projects with changing or unclear requirements and where customer collaboration is crucial.

- Phases: No fixed phases; work is organized into iterations (Sprints) with planning, coding, testing, and review activities within each iteration.

7.6 BIG BANG MODEL

- Description: The Big Bang model is an informal and chaotic approach where there is no specific process or phases defined. Developers work on the project without a structured plan.

- Use: It is typically used for small projects or proof-of-concept development.

- Phases: Not defined; often lacks a formal process.

7.7 INCREMENTAL MODEL

- Description: The Incremental model divides the project into smaller, manageable parts called increments. Each increment represents a portion of the functionality, and new increments are added iteratively.

- Use: Useful for large projects with long development cycles where partial functionality can be delivered early.

- Phases: Each increment follows phases similar to the Waterfall model.

8. SDLC FOR NEONTECH

8.1 HOW AGILE MODEL USED IN WEBSITE?

The Agile model is commonly used in website development to promote flexibility, collaboration, and continuous improvement. Here's how Agile principles and practices can be applied in the context of website development:

1. Product Vision: Clearly define the vision and goals for the website. Understand the target audience, business objectives, and key features.
2. User Stories: Break down the features into user stories. User stories are concise descriptions of functionality from the user's perspective.
3. Product Backlog: Maintain a prioritized product backlog that lists all the features, enhancements, and bug fixes.
4. Sprints Planning: At the beginning of each sprint, select a set of user stories from the backlog based on priority and feasibility.
5. Time-Boxed Development: Work on these selected user stories during the sprint, which is a time-boxed period (commonly two weeks). At the end of the sprint, deliver a potentially shippable increment of the website.
6. Scrum Roles: Facilitate the Agile process, remove impediments, and ensure the team is following Agile principles.
7. Product Owner: Represents the stakeholders and ensures the development team is working on the highest-priority features.
8. Daily Stand-Ups: Conduct daily stand-up meetings to discuss progress, challenges, and plans. This fosters communication and collaboration among team members.

9. Continuous Integration and Continuous Deployment (CI/CD): Implement CI/CD pipelines to automate the build, test, and deployment processes. This ensures that changes are integrated and tested frequently.
10. Feedback and Iteration: Gather feedback from stakeholders, users, and team members regularly. Use this feedback to make adjustments and improvements in subsequent sprints.
11. Retrospectives: Hold regular retrospectives at the end of each sprint to reflect on what went well, what could be improved, and how to enhance the development process.
12. Adaptable Planning: Be adaptable to changes in requirements. Agile allows for reprioritization based on evolving business needs or user feedback.
13. Cross-Functional Teams: Form cross-functional teams that include developers, testers, designers, and other necessary roles. This promotes collaboration and the ability to deliver a complete, high-quality product.
14. Prototyping and Minimal Viable Product (MVP): Use prototyping and develop a Minimal Viable Product (MVP) early in the project to get a basic version of the website into the hands of users quickly.
15. User Acceptance Testing (UAT): Involve users in the testing process. Conduct user acceptance testing to ensure that the delivered features meet user expectations.
16. 13. Documentation: Agile values working software over comprehensive documentation. However, essential documentation, such as user stories and acceptance criteria, is maintained to ensure a shared understanding among team members.
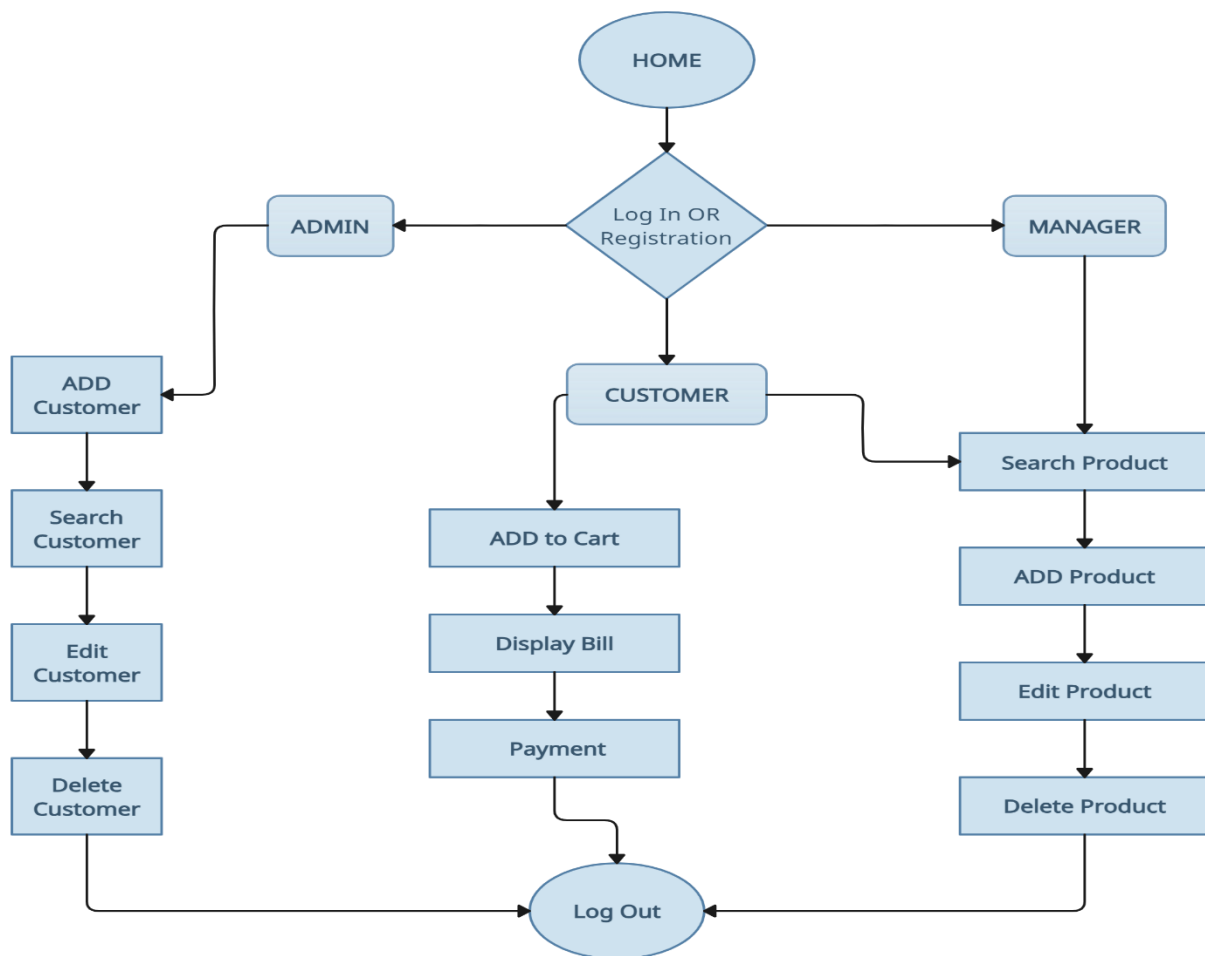
8.2 WHY AGILE MODEL IS GOOD FOR WEBSITE?

The Agile model is well-suited for website development for several reasons, and its advantages contribute to more effective, adaptable, and successful projects. Here are some reasons why the Agile model is good for website development:
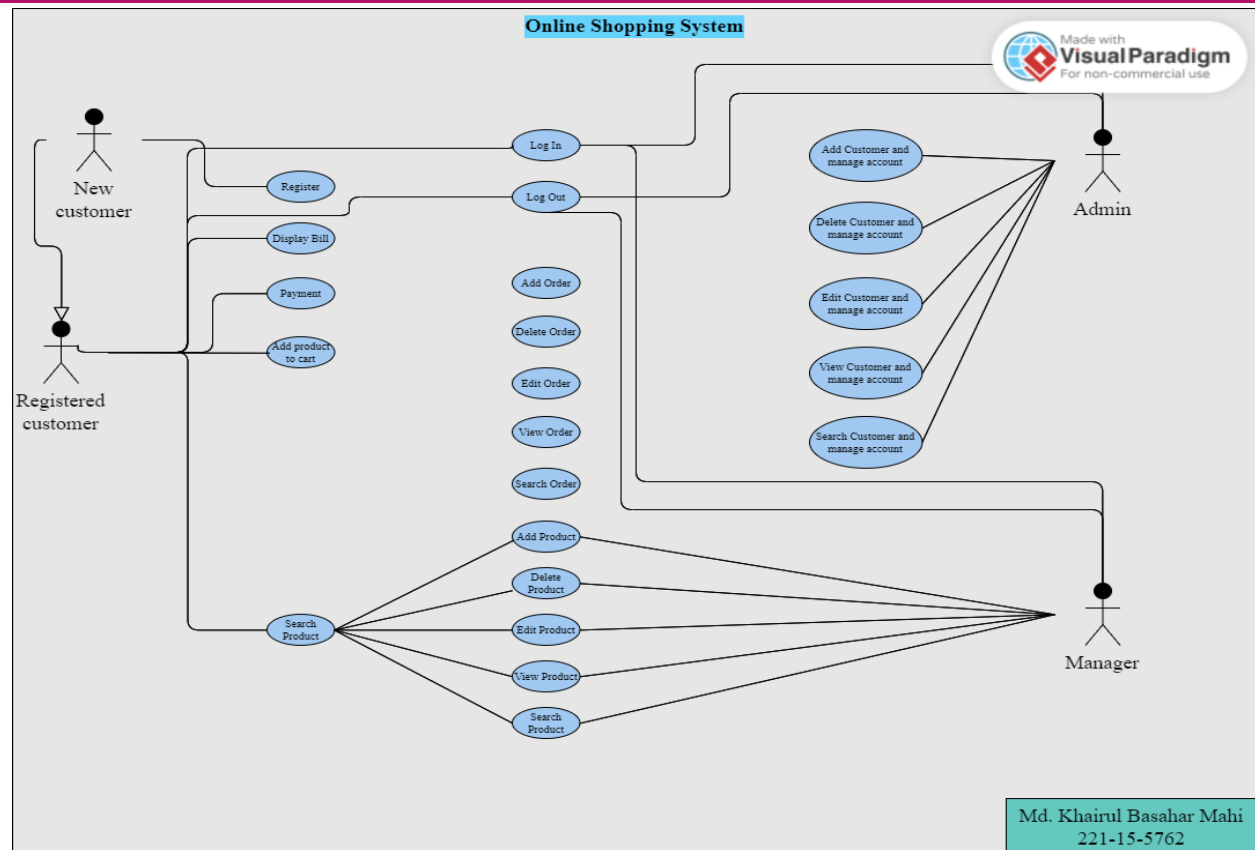
1. Frequent Delivery of Value: Agile emphasizes delivering a Minimum Viable Product (MVP) quickly and then iterating on it. This allows for the rapid release of features and improvements, providing value to users sooner.
2. Adaptability to Change: Websites often require frequent updates and adjustments based on changing market trends, user feedback, and emerging technologies. Agile iterative and incremental approach makes it easier to adapt to these changes, ensuring that the website remains relevant.
3. Continuous User Feedback: Agile promotes regular interactions with users and stakeholders, allowing for continuous feedback. This ensures that the website aligns with user expectations, and adjustments can be made throughout the development process.
4. Improved Collaboration: Agile encourages close collaboration between cross-functional teams, including developers, designers, testers, and product owners. This collaboration fosters a shared understanding of goals and requirements, leading to a more cohesive and well-rounded website.
5. Reduced Time-to-Market: The iterative nature of Agile allows for the frequent release of increments. This, combined with the ability to reprioritize features, contributes to a reduced time-to-market. Websites can quickly respond to market demands and user needs.

6. Customer-Centric Approach: Agile places a strong emphasis on understanding and delivering what the customer truly values. This customer-centric approach is vital for websites, where user satisfaction and engagement are critical to success.
7. Transparency and Visibility: Agile methodologies provide transparency into the development process. Regular meetings, such as daily stand-ups and sprint reviews, offer visibility into progress, challenges, and upcoming work, fostering a culture of openness.
8. Increased Quality Through Testing: Agile includes testing as an integral part of the development process. Continuous testing, including automated testing, helps ensure the quality of the website. The emphasis on regular testing contributes to the early detection and resolution of issues.
9. Iterative Design and Development: Websites often require design improvements and adjustments. Agile iterative nature allows for continuous refinement of design elements, ensuring that the website's user interface and user experience evolve based on user feedback and changing requirements.
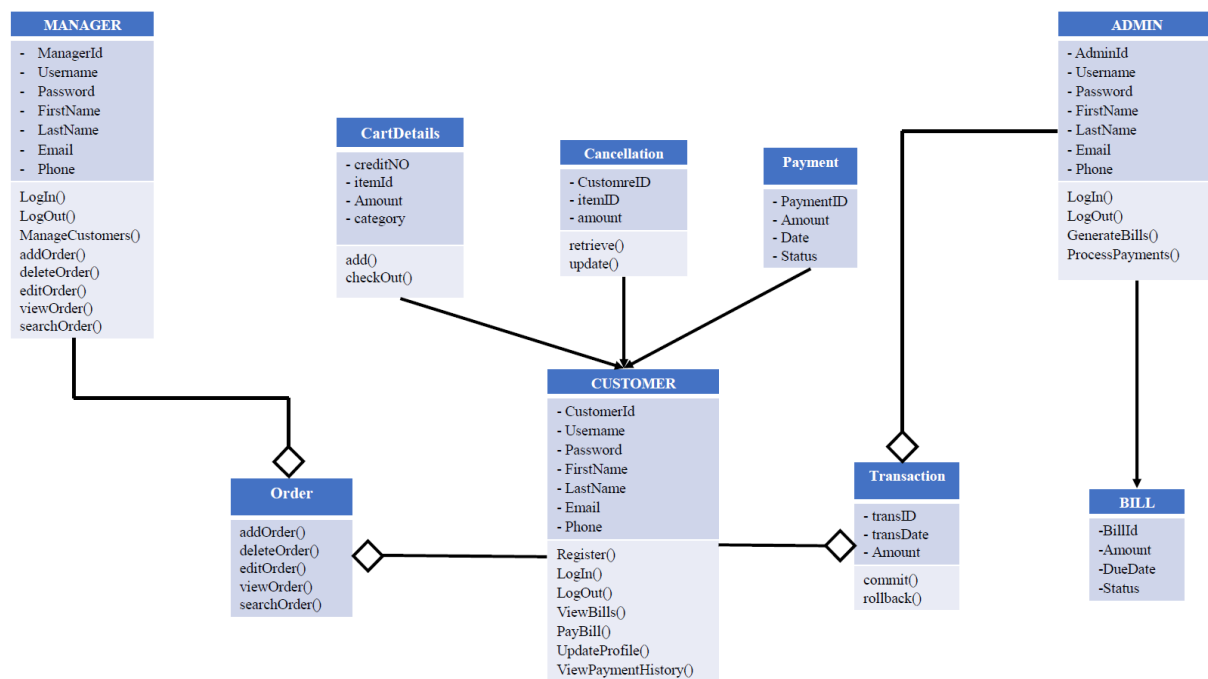
9. FLOW CHART

10. USE CASE

Online Shopping System

Made with VisualParadigm
For non-commercial use

New customer

Log In
Register
Display Bill
Payment
Add product to cart

Log Out
Add Order
Delete Order
Edit Order
View Order
Search Order

Registered customer

Add Customer and manage account
Delete Customer and manage account
Edit Customer and manage account
View Customer and manage account
Search Customer and manage account

Admin

Search Product

Add Product
Delete Product
Edit Product
View Product
Search Product

Manager

Md. Khairul Basahar Mahi
221-15-5762

11. UML DIAGRAM

MANAGER
- ManagerId
- Username
- Password
- FirstName
- LastName
- Email
- Phone

LogIn()
LogOut()
ManageCustomers()
addOrder()
deleteOrder()
editOrder()
viewOrder()
searchOrder()

CartDetails
- creditNO
- itemId
- Amount
- category

add()
checkOut()

Cancellation
- CustomreID
- itemID
- amount

retrieve()
update()

Payment
- PaymentID
- Amount
- Date
- Status

ADMIN
- AdminId
- Username
- Password
- FirstName
- LastName
- Email
- Phone

LogIn()
LogOut()
GenerateBills()
ProcessPayments()

CUSTOMER
- CustomerId
- Username
- Password
- FirstName
- LastName
- Email
- Phone

Register()
LogIn()
LogOut()
ViewBills()
PayBill()
UpdateProfile()
ViewPaymentHistory()

Transaction
- transID
- transDate
- Amount

commit()
rollback()

BILL
-BillId
-Amount
-DueDate
-Status

Order
addOrder()
deleteOrder()
editOrder()
viewOrder()
searchOrder()

12. DEPLOYMENT

The Application will completely base on the following requirement which is given by your company and this Application is cover those features which is right down. For development time developer only focus on the feature. If client want more feature, then he has to pay base on new feature.

16.TESTING

16.1 PROPOSED TESTING APPROACH:

For a tech online shopping website, we need a comprehensive and layered strategy that encompasses various testing levels, types, and techniques. This approach should align with the website's specific requirements, functionalities and targeted audience.

We proposed **End-to-End Testing with Agile Methodology.**

Compared to traditional linear testing approaches, this comprehensive strategy offers several advantages:

- Early defect detection: By incorporating testing activities throughout the development lifecycle, defects can be identified and rectified early, reducing the cost and complexity of fixing them later.
- Improved quality and user experience: By thoroughly testing the website at various levels and using diverse techniques, the overall quality and user experience are enhanced.
- Reduced risk of production issues: Proactive testing mitigates the risk of critical defects reaching production, ensuring a smooth and reliable user experience.
- Perform comprehensive testing covering all aspects of the system, from user interfaces to back-end processes.
- Use Agile methodology for flexibility and quicker adaptation to changes.

16.2 DISCUSSION OF TESTING LEVELS

Testing levels refer to the scope and depth of testing performed at different stages of the development lifecycle. For a tech online shopping website, the following testing levels are crucial:

a) Unit Testing:
- Focus: Individual components, such as product search algorithms or payment processing functions.
b) Integration Testing:
- Focus: Interactions between different modules (e.g., cart integration with payment systems).
c) System Testing:
- Focus: Overall system functionality, including user interfaces, databases, and server interactions.
d) User Acceptance Testing (UAT):
- Focus: Ensure the website meets user expectations, including browsing, selecting products, and making purchases.

16.3 TESTING TYPES, TECHNIQUES, AND TACTICS:

i. **Testing Types:**
- Functional Testing: Test product search, order placement, payment processing, and other core functions.
- Performance Testing: Assess website speed, responsiveness, and scalability during peak loads.
- Security Testing: Identify and address vulnerabilities to protect user data.
- Usability Testing: Evaluate the user-friendliness of the website.
- Compatibility Testing: Ensure the website works across various browsers and devices.

ii. **Testing Techniques:**
- Black Box Testing: Evaluate the website's functionality without knowledge of internal code.
- Load Testing: Simulate expected user loads to ensure the website can handle traffic.
- Security Scanning: Use tools to scan for common security vulnerabilities.

iii. **Testing Tactics:**
- Regression Testing: Ensure that new updates do not negatively impact existing features.
- Exploratory Testing: Simulate real user scenarios to uncover unexpected issues.
- User Flow Testing: Test typical user journeys from product search to order completion.

16.4 PROPOSED TESTING PROCESS

a. Requirements Analysis: Understand and document project requirements, including user stories and acceptance criteria.
b. Test Planning: Develop a detailed test plan covering all testing phases, including resource allocation and schedule.
c. Test Design: Create comprehensive test cases covering functional, performance, and security aspects.
d. Test Execution: Execute test cases manually or use automated tools for efficiency.
e. Defect Reporting: Document and prioritize defects found during testing.
f. User Acceptance Testing (UAT): Involve end-users to validate the system against their requirements.
g. Deployment: After successful testing, deploy the website to production.

16.5 MEASUREMENT IN SOFTWARE TESTING (HIERARCHY OF TESTING DIFFICULTY)

a. Hierarchy of Testing Difficulty:
- Unit Testing: Generally, less complex as it focuses on individual components.
- Integration Testing: Moderate complexity, ensuring that components work well together.
- System Testing: Higher complexity, testing the entire system's functionality.
- User Acceptance Testing (UAT): Can be complex as it involves end-users' perspectives.

b. Metrics:
- Conversion Rate: Measure the percentage of website visitors who make a purchase.
- Average Session Duration: Evaluate the average time users spend on the website.
- Error Rates: Monitor the frequency of errors encountered by users during their interactions.

c. Continuous Improvement: Regularly review and enhance testing processes based on feedback, metrics, and evolving requirements.

17. SUPPORT

- 24/7 support based on payment.

18. PRICING

Our fee for seeing the project through from start to completion will be ***Thirty-Four Thousand Nine Hundred Ninety-Nine Taka*** (34,999 Tk).

19. PAYMENT TERMS

We propose the following payment terms:

10% (10%)

Paid on acceptance of this proposal.

40% (50%)

Paid on signing of our application development agreement.

25% (75%)

Paid at 70% Application Demonstration.

25% (100%)

Paid at completion the Application.

20. RESPONSIBILITY:

This Application Ordered by **MD. TARIKUR RAHMAN**, Managing Director of UCC, all responsibility goes on him.

This Application will Developed by Shrikanta Paul and Md. Khairul Bashar on a project of Neon-Tech Development Group supervised by Md. Khairul Bashar.

21. CONTACT US

You can get in touch with us in any of the below ways:

By Phone:
+8801765771733
Md. Khairul Bashar
By Email
khairulbasharmahi@gmail.com

On our website
www.neontech.com
By post
Room-141, 4 No, House-building
Eye Hospital,
Uttara, Dhaka-1330

We look forward to hearing from you soon!

Agreement Signed By:

……………………………   ……………………………

Client Signature   Authority Signature
MD. TARIKUR RAHMAN   Md. Khairul Bashar
Managing Director   Assistant Head
UCC   Nean-Tech