|  |
| --- |
| **Software Requirements Specification** |
| **Telemarketing Website and Web Application System** |
|  |
|  |
|  |
| **Bony** |
| **11/4/2023** |
|  |

Md. Kholilur Rahman Rabby

Managing Director

Boss Telemarketing

YKSG-2,Ground Floor.

DSC,Savar,Dhaka

Dear Md. Kholilur Rahman Rabby,

Re: Application Agreement for Telemarketing Website and Web Application System.

At Young’s IT, 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 will design a “Telemarketing Website and Web Application System” for you, in this agreement paper we shall right each and every part and all possible feature’s which covered our Application. So please read carefully and if it satisfy you then sign and confirm the agreement.

Yours Truly,

Bony

Head

Application Development Brach

Young’s IT

**1.Project**

Telemarketing Website and Web Application System

**2. 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 consist a website and a web application, Website is fully information based and the web application has two modules with I-cloud features.

**3. Obstacles**

1. Platform
2. Manage Storage File System

**4. User Features in Details:**

Telemarketing Website and Web Application System Has two part, They Are:

* **Website** 
  + Home
  + About Us
  + Previous projects (On Click A Project View Project Details)
  + Contact
  + Login
* **Web Application System**
  + **User**
    - Login
    - Payment Status
    - Payment History
    - Download Receipt
  + **Admin**
    - Create User
    - Submit User Payment
    - See User Payment Status
    - Payment History
    - Generated Receipt

**5. Technology Require (Application and Hardware)**

1. **Framework:** Slim.
2. **Database:** MySQL.
3. **Design:** Standard.
4. **Coding Architecture:** OOP.
5. **Security:** Standard.

**6. Preliminary Schedule**

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Reporting** | **Time** |
| Analysis | Submit The Design | 3 days |
| Requirements Collection |  | 5 days |
| Development | Review The Work | 30 days |
| Testing |  | 6 days |
| Deployment | Review Final Work | 3 days |
| Delivery | Live On Server | 3 days |

**7. Pricing**

Our fee for seeing the project through from start to completion will be fifty Thousand Taka only (50,000Tk).

**8. Payment Terms**

We propose the following payment terms:

20% (20%)🡪 Paid on acceptance of this proposal.

25% (45%)🡪 Paid on signing of our Application development agreement.

25% (70%)🡪Paid at 70% completion of Application Demonstration.

30% (100%)🡪Paid at 100% completion of the Application.

**9. Deployment**

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

**10. Software Development Life Cycle (SDLC)**

In order to develop software or a website we use many types of SDLC models. Such as: Waterfall Model, RAD Model, Spiral Model, V-Model, Incremental Model, Agile Model, Iterative Model, Big Bang Model. All of these models have some advantage and disadvantage.

Those advantage and disadvantages are:

**Waterfall Model**

**Advantage:**

1. This model is simple to implement also the number of resources that are required for it is minimal.
2. The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
3. The start and end points for each phase is fixed, which makes it easy to cover progress.
4. The release date for the complete product, as well as its final cost, can be determined before development.
5. It gives easy to control and clarity for the customer due to a strict reporting system.

**Disadvantage:**

1. In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
2. This model cannot accept the changes in requirements during development.
3. It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.
4. Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

**RAD Model**

**Advantage:**

1. This model is flexible for change.
2. In this model, changes are adoptable.
3. Each phase in RAD brings highest priority functionality to the customer.
4. It reduced development time.
5. It increases the reusability of features.

**Disadvantage:**

1. It required highly skilled designers.
2. All application is not compatible with RAD.
3. For smaller projects, we cannot use the RAD model.
4. On the high technical risk, it's not suitable.
5. Required user involvement.

**Spiral Model**

**Advantage:**

1. High amount of risk analysis
2. Useful for large and mission-critical projects.

**Disadvantage:**

1. Can be a costly model to use.
2. Risk analysis needed highly particular expertise
3. Doesn't work well for smaller projects.

**V-Model**

**Advantage:**

1. Easy to Understand.
2. Testing Methods like planning, test designing happens well before coding.
3. This saves a lot of time. Hence a higher chance of success over the waterfall model.
4. Avoids the downward flow of the defects.
5. Works well for small plans where requirements are easily understood.

**Disadvantage:**

1. Very rigid and least flexible.
2. Not a good for a complex project.
3. Software is developed during the implementation stage, so no early prototypes of the software are produced.
4. If any changes happen in the midway, then the test documents along with the required documents, has to be updated.

**Incremental Model**

**Advantage:**

1. Errors are easy to be recognized.
2. Easier to test and debug
3. More flexible.
4. Simple to manage risk because it handled during its iteration.
5. The Client gets important functionality early.

**Disadvantage:**

1. Need for good planning
2. Total Cost is high.
3. Well defined module interfaces are needed.

**Agile Model**

**Advantage:**

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfils the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

**Disadvantage:**

1. Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
2. Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

**Iterative Model**

**Advantage:**

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

**Disadvantage:**

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

**Big Bang Model**

**Advantage:**

1. There is no planning required.
2. Simple Model.
3. Few resources required.
4. Easy to manage.
5. Flexible for developers.

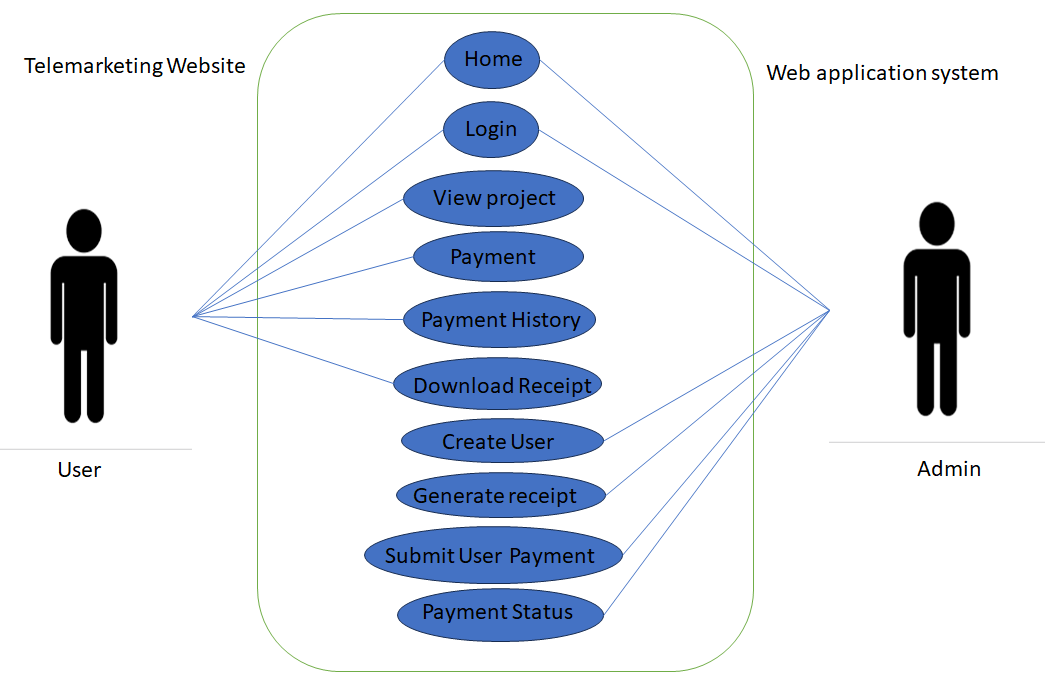
**Disadvantage:**

1. There are high risk and uncertainty.
2. Not acceptable for a large project.
3. If requirements are not clear that can cause very expensive.

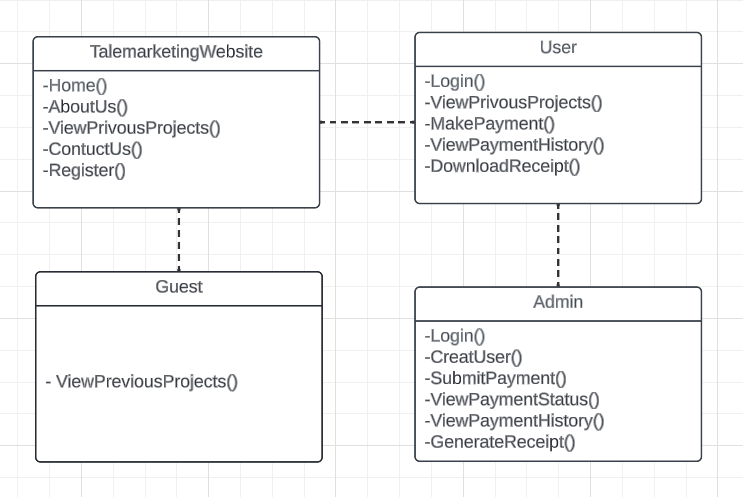
After analyzing all these advantages and disadvantages of these most popular SDLC models we think it would be better to use “Agile Model” rather than using any other model in order to make your website and web application system.

**11. Use case diagram & UML**

**Use case diagram**

****

**UML**

****

**12. Web application Test plan , Test Case:**

**Web Application Testing:**

Web Application Testing is checking web application for potential bugs before it is made live and is accessible to general public. For our project we will be using Manual Testing, because compared to other testing methods it can detect more bug and system failure.

* **Web Application Testing levels:**

Web application Testing checks for: functionality, usability, interface, compatibility, performance and the security of the web application.

**Functionality Testing:**

The primary objective of functional testing is to ensure that all of the functionalities of a web application function smoothly without issues. Functional testing in a web application may include verifying whether all links are operating correctly, testing cookies, reviewing forms on all web-pages, evaluating database security, and validating CSS or HTML.

**Usability Testing:**

Usability testing is a type of testing conducted to evaluate the user-friendliness, effectiveness, efficiency, and overall user experience of a web application. The primary goal of usability testing is to ensure that the web application is intuitive, easy to navigate, and meets the expectations of its intended users. This type of testing focuses on how well users can interact with the application and accomplish specific tasks.

**Interface Testing:**

Critical components of a web app are the web server interface, database server interface, and application server interface. Interface Testing validates that all components of the web application are well-connected. Testers use this testing method to determine whether the interaction between these servers is carried out effectively.

**Compatibility Testing:**

Compatibility is a critical factor to be considered while testing a web application. Compatibility testing determines the compatibility of a web application with operating systems, browsers, mobile browsing, and other printing options.

**Performance Testing:**

Performance testing enables tester to determine a web application's performance in various scenarios. It includes stress testing, load testing, and scalability testing. Typically, this testing method evaluates a website's functionality across different hardware platforms, operating systems.

**Security Testing:**

Security testing is a critical web application testing method as it determines whether data modifications are tolerable or not. It primarily includes various activities such as verifying SSL, checking the CAPTCHA for automated script logins, and determining whether web files or directories can be accessed directly.

* **Web Application Testing Types:**

Web application testing involves various types of testing to ensure the reliability, security, performance, and functionality of the application. Here are some common types of web application testing:

**Functional Testing:**

**Unit Testing:** Testing individual components or functions of the application in isolation.

**Integration Testing:** Verifying that different components/modules work together as expected.

**System Testing:** Evaluating the complete and integrated web application to ensure it meets the specified requirements.

**Non-functional Testing:**

**Performance Testing:** Assessing the responsiveness, speed, and overall performance of the application under various conditions.

**Load Testing:** Evaluating the application's ability to handle a specific load or number of concurrent users.

**Stress Testing:** Subjecting the application to conditions that exceed normal usage to identify its breaking points.

**Scalability Testing:** Assessing how well the application scales as user load increases.

**Usability Testing:** Evaluating the user interface, user experience, and overall usability of the application.

**Compatibility Testing:** Ensuring the application works consistently across different browsers, devices, and operating systems.

**Reliability Testing:** Testing the application's ability to perform consistently and reliably under normal and adverse conditions.

**Security Testing:**

**Vulnerability Assessment**: Identifying and addressing potential vulnerabilities in the application's code and configurations.

**Penetration Testing:** Simulating real-world attacks to uncover security weaknesses in the application.

**Authentication and Authorization Testing:** Verifying the effectiveness of user authentication and access control mechanisms.

**Database Testing:**

**Data Integrity Testing:** Ensuring the accuracy and consistency of data stored in the database. **Data Migration Testing:** Verifying the successful transfer of data between databases or systems.

**Database Performance Testing:** Assessing the performance of database queries and transactions.

**Regression Testing:**

- Ensuring that new code changes do not negatively impact existing features and functionalities.

- Re-executing a subset of test cases to validate that the application remains stable.

**User Acceptance Testing (UAT):**

- Involving end-users to validate whether the application meets their expectations and business requirements.

- Conducted in a controlled environment before the application goes live.

**Cross-browser Testing:**

- Verifying that the web application works correctly and consistently across different web browsers.

**Mobile Application Testing:**

- Ensuring the functionality and user experience of the web application on mobile devices.

- Testing responsiveness, performance, and device-specific features.

**Exploratory Testing:**

- Investigating the application without predefined test cases to discover defects and usability issues.

**Localization and Internationalization Testing:**

- Verifying that the application supports different languages and regions.

- Ensuring that cultural and regional requirements are met.

**API Testing:**

- Evaluating the functionality, reliability, and security of APIs used in the web application.

These testing types can be applied throughout the development life cycle to identify and address issues at different stages, ensuring the overall quality of the web application. The specific testing strategy depends on the project requirements, goals, and constraints.

* **Manual Testing Process:**

Manual testing for a web application involves a tester manually executing test cases without using any automation tools. The process typically involves the following steps:

**Understanding Requirements:**

- Obtain a clear understanding of the project requirements, specifications, and user stories.

- Collaborate with stakeholders such as developers, product managers, and business analysts to gather information about the application.

**Test Planning:**

- Develop a test plan outlining the testing approach, scope, resources, schedule, and deliverables.

- Identify test scenarios, test cases, and test data based on the requirements.

**Environment Setup:**

- Set up the testing environment, including the necessary hardware, software, and network configurations.

- Ensure that the test environment closely mimics the production environment.

**Test Case Design:**

- Design detailed test cases based on the specified test scenarios.

- Test cases should cover positive and negative scenarios, boundary conditions, and various user interactions.

**Execution of Test Cases:**

- Manually execute the test cases by interacting with the web application's user interface.

- Verify that the application behaves as expected and meets the specified requirements.

- Document the test results, including any defects or issues encountered during testing.

**Defect Reporting:**

- If any defects are found during testing, document them in a defect tracking system.

- Include detailed information about the defect, steps to reproduce, and any relevant screenshots or logs.

**Regression Testing:**

- After developers fix reported defects, perform regression testing to ensure that the changes do not introduce new issues.

- Re-execute relevant test cases to verify the overall stability of the application.

**Exploratory Testing:**

- Conduct exploratory testing to simulate real-world usage scenarios and discover unexpected issues.

- Explore different paths within the application to identify usability problems or unanticipated behavior.

**Ad Hoc Testing:**

- Perform ad hoc testing to validate the application's functionality without following a predefined test script.

- Testers may use their experience and intuition to identify potential areas of concern.

**Usability Testing:**

- Evaluate the user interface, navigation, and overall user experience to ensure that the application is user-friendly.

- Provide feedback on any usability issues or suggestions for improvement.

**Performance Testing:**

- Evaluate the performance of the web application by manually testing its responsiveness, load times, and scalability.

- Identify and report any performance-related issues.

**Documentation:**

- Maintain comprehensive documentation, including test plans, test cases, test data, and test results.

- Ensure that all testing activities are well-documented for future reference.

Manual testing is essential for its flexibility, especially in scenarios where test cases are complex, require human intuition, or when the application's user interface needs careful evaluation.

* **Measurement in Web App testing (Hierarchy of testing):**

While testing a web application a tester has to face many difficulties. Such as:

**Interaction with Firewalls:**

Challenges in connecting with firewalls arise because a port or a firewall can block a web app due to issues with security certificate compliance. Thus, it is critical to test the application across multiple firewalls to avoid such scenarios.

**Validation of web services:**

Modern web applications rely heavily on web service layers such as XML/SOAP or JSON/REST for exchanging data between applications or systems. Hence, the need for verifying these web services has increased significantly. However, web testing platforms are not equipped to handle them. As a result, testing web services is a significant problem for the QA teams.

**Consistency across browsers:**

Customers always prefer scalable and highly interactive web applications. If a customer encounters inconsistencies while browsing across browsers, it can adversely impact the enterprise's brand image and growth. Thus, developers and QA testers must track the app's scalability and interactivity across browsers and hardware configurations while conducting usability testing.

**Addressing performance issues:**

Users abandon any slow-loading web app, which eventually affects the company brand and revenue. Thus, to overcome this obstacle, developers should identify the factors affecting performance testing, such as scalability of the app's features, compatibility, and integration limitations.

**Securing web app against data breaches:**

There has been a constant rise in the number of cyber threats that can leak users' sensitive data. It is essential to monitor security testing processes to avoid loss of information and data breaches. It will help prevent DDoS attacks and other cyber threats. In case of any issue detection, testers can fix those at the earliest.

**13. Technological Requirement:**

In order to make a web application there are some technological requirements. Such as:

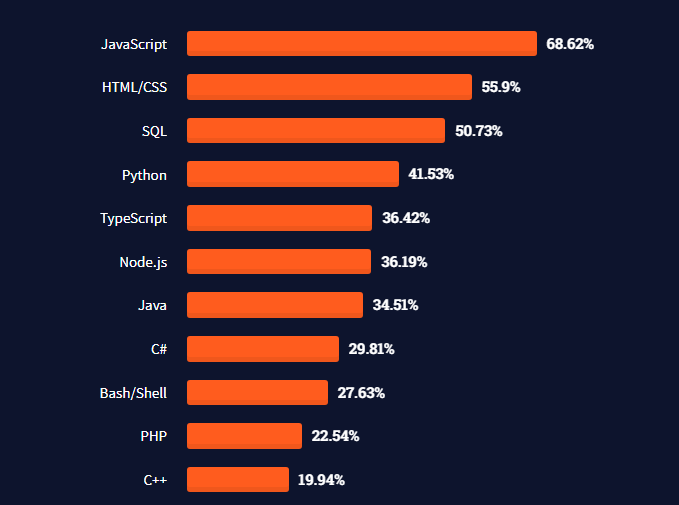
1. Software
2. Hardware

* **Software**

In software requirements developers has to choose some things. Such as:

**Font-end Language:**

There is a lot of popular programming language for making a web application, and based on their popularity they are vastly used. We can see in the below table the popularity of those languages.



For your web application we will be using HTML/CSS, because it would be much more fitting for your web application.

**Back-end Language:**

Since your web application is medium-size we will be using Python as its Back-end Language.

**Database:**

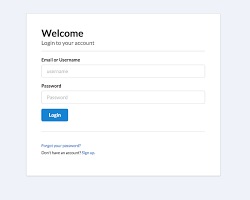
There are a lot of databases. Such as: MySQL, **PostgreSQL, MongoDB, MariaDB, SQLite excreta. We will be MySQL for your web application database.**

**Security:**

Securing a medium-sized web application with MySQL as the database is crucial for protecting sensitive data and maintaining the integrity of the application. Implementing a comprehensive security strategy that encompasses user authentication, data encryption, network protection, and vulnerability management is essential to safeguard application against potential threats.

**User Authentication**

1. Strong Passwords: Enforce strong password policies, requiring complex passwords with a combination of upper and lowercase letters, numbers, and symbols. Encourage the use of unique passwords for different accounts and avoid password reuse.
2. Multi-Factor Authentication (MFA): Implement MFA to add an extra layer of security beyond passwords. MFA can include methods like one-time passwords (OTPs) sent to mobile phones or email, or security tokens.
3. Role-Based Access Control (RBAC): Grant users only the minimum privileges necessary to perform their tasks. RBAC restricts access to specific data and functions based on user roles, preventing unauthorized access to sensitive information.

[](https://tilomitra.medium.com/how-to-create-a-node-web-app-with-user-authentication-in-under-10-minutes-47041f96f6ca)

**Data Encryption**

1. Encrypt Stored Data: Encrypt sensitive data stored in the MySQL database using strong encryption algorithms like AES or RSA. Encryption protects data from unauthorized access even if the database is breached.
2. Encrypt Data in Transit: Encrypt data transmitted between the application server and the database server using protocols like Transport Layer Security (TLS) or Secure Sockets Layer (SSL). This prevents eavesdropping and data tampering during transmission.

[](https://www.netsolutions.com/insights/how-to-protect-data-in-mobile-web-apps-using-encryption/)

**Network Protection**

1. Firewalls: Implement a firewall to restrict unauthorized access to the database server from the outside world. Configure the firewall to allow only trusted connections and block access from suspicious sources.
2. Intrusion Detection/Prevention Systems (IDS/IPS): Deploy IDS/IPS solutions to monitor network traffic for malicious activity and potential intrusions. IDS systems detect attacks, while IPS systems can actively block them.
3. Web Application Firewalls (WAFs): Utilize WAFs to protect the application from common web attacks like SQL injection and cross-site scripting (XSS). WAFs filter incoming requests and block malicious traffic.

[](https://avinetworks.com/what-is-a-web-application-firewall/)

**Vulnerability Management**

1. Regular Security Audits: Conduct regular security audits to identify and address vulnerabilities in the application and database. Vulnerabilities can be exploited by attackers to gain unauthorized access or compromise data.
2. Patch Management: Promptly apply security patches and updates for the operating system, application software, and database server. Patches fix vulnerabilities and address known security flaws.
3. Security Awareness Training: Provide security awareness training to developers and employees to educate them about cybersecurity threats and best practices. This helps prevent human error and social engineering attacks.



By implementing these security measures, you can significantly enhance the security posture of your medium-sized web application using MySQL as the database, protecting sensitive data, maintaining application integrity, and safeguarding against potential cyber threats.

**Model-View-Controller:**

Model-View-Controller (MVC) is a software design pattern commonly used in web development. It separates the application into three interconnected parts: the model, the view, and the controller. This modular approach promotes code organization, maintainability, and testability.

**Model:** The model represents the application's data and business logic. It encapsulates the data structures and operations that manipulate the data. The model interacts with the database to retrieve, store, and update data.

**View:** The view is responsible for rendering the application's user interface (UI). It transforms the model's data into a form that can be displayed to the user. The view typically uses HTML, CSS, and JavaScript to create the UI components.

**Controller:** The controller acts as an intermediary between the model and the view. It handles user interactions, updates the model accordingly, and instructs the view to render the updated data. The controller manages the overall flow of the application.

For your web application we will be using MVC V5.

MVC V5 refers to the fifth major version of the ASP.NET MVC framework, an open-source web application framework for building web applications on the .NET platform. It implements the MVC pattern and provides a variety of features to simplify web development.

MVC V5 introduces several improvements over previous versions, including:

* Enhanced routing capabilities for handling complex URL patterns
* Improved asynchronous controller actions for efficient handling of requests
* Support for attribute-based routing for a more declarative approach to routing
* Integration with Razor Pages, a lightweight alternative to MVC for simple pages
* Enhanced dependency injection framework for managing object dependencies

MVC V5 is a powerful and versatile framework for building web applications on the .NET platform. Its use of the MVC pattern promotes code organization, maintainability, and testability, making it a popular choice for web developers.

* **Hardware**

While developing a web application developers need to keep an eye on the Memory size and the Bandwidth of the web application.

**Memory size:** Based on the amount of data that needed to be store, the number of users, the complexity of the application, **the efficiency of the application's code** the memory size can differ. But for now we will be providing you with a 25 gigabyte memory space. In future if needed the memory size can be upgraded.

**Bandwidth:** Bandwidth is the maximum amount of data that can be transferred between a web application and its users in a given period of time. It is typically measured in bits per second or megabits per second.

The bandwidth of a web application is determined by several factors, including:

* The size of the application's pages and other content
* The number of users accessing the application
* The frequency of user interactions with the application
* The type of content being served, such as text, images, or video

A web application with a lot of large pages, a high number of users, and frequent user interactions will require more bandwidth than a web application with smaller pages, fewer users, and less frequent user interactions. Additionally, web applications that serve a lot of video content will require more bandwidth than web applications that serve mostly text and images.

To ensure that your web application has sufficient bandwidth, we will be monitoring your application's bandwidth usage and traffic patterns. We will also be using a content delivery network to distribute your application's content across multiple servers around the world.

But for starting we will provide you with a Bandwidth of 20 megabits per second (20 Mbps). You can also change in future if needed.

**14. Maintenance Plan**

The system will be maintained on a regular basis to ensure that it is functioning properly. The tasks that will be performed as part of the maintenance plan are:

1. Bug fixes
2. Security updates
3. New features

**15. Responsibility:**

This Application Ordered by Md. Kholilur Rahman Rabby, Managing Director of Boss Telemarketing, All responsibility goes on him.

**16. Contact Us**

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

**By Phone:**

+8801719019635

Khorshed Alom

**By E-mail**

Kr505599@gmail.com

On our website

<www.young.it>

**By post :**

Room-25,Mosjid Market,

Madrasa Mor,

Natore,Rajshahi.

**Agreement Signed By:**

|  |  |  |
| --- | --- | --- |
| ……………………………  Client Signature  Md. Kholilur Rahman Rabby  Managing Director  Boss Telemarketing | …………………………..  Order Provider Signature  Officer  Young’s IT | ……………………………  Authority Signature  Md. Khorshed Alom  Managing Director (MD)  Young’s IT |