

SAC-Scripture And Canticle 3.0

A mini project report submitted by

CHRISTOPHER PAUL L (URK19CS1173)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Dr. BELFIN R V, Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Declared as Deemed-to-be-under Sec-3 of the UGC Act, 1956)

Karunya Nagar, Coimbatore - 641 114. INDIA

March 2022



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION

AICTE Approved & NAAC Accredited

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled, “**SAC-Scripture And Canticle 3.0**” is a bonafide record of Mini Project work done during the even semester of the academic year 2021-2022 by

CHRISTOPHER PAUL L (Reg. No: URK19CS1173)

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Karunya Institute of Technology and Sciences.

Submitted for the Viva Voce held on 23RD March,2022.

Signature of the Guide

ACKNOWLEDGEMENT

First and foremost, I praise and thank ALMIGHTY GOD whose blessings have bestowed in me the will power and confidence to carry out my project.

I am grateful to our beloved founders Late. **Dr. D.G.S. Dhinakaran, C.A.I.I.B, Ph.D** and **Dr. Paul Dhinakaran, M.B.A, Ph.D**, for their love and always remembering us in their prayers.

I extend my thanks to our Vice Chancellor **Dr. P. Mannar Jawahar, Ph.D** and our Registrar **Dr. Elijah Blessing, M.E., Ph.D**, for giving me this opportunity to do the project.

I would like to thank **Dr. Prince Arulraj, M.E., Ph.D.**, Dean, School of Engineering and Technology for his direction and invaluable support to complete the same.

I would like to place my heart-felt thanks and gratitude to **Dr. J. Immanuel John Raja, M.E., Ph.D.**, Head of the Department, Computer Science and Engineering for his encouragement and guidance.

I feel it is a pleasure to be indebted to, **Dr. J. Andrew, M.E, (Ph.D.)**, Assistant Professor, Department of Computer Science and Engineering and **Dr.Belfin R V M.E., Ph.D.**, for their invaluable support, advice and encouragement.

I also thank all the staff members of the Department for extending their helping hands to make this project a successful one.

I would also like to thank all my friends and my parents who have prayed and helped me during the project work.

Abstract

The Christian church is almost millennia old, starting with the beginning of the church on the time of Pentecost. Although there are sure constants in Christian worship, which includes the preaching of the phrase and administering the sacraments, worship additionally displays the traditions and context wherein people live. The fashion of music, the aesthetics of worship, the language spoken, and church structure have all been fashioned, to at least one quantity or another, with the aid of using the encircling culture. Technology, as a part of culture, has additionally fashioned the church in giant ways. To recognize how era shapes the church one ought to understand that era isn't always pretty much the nuts and bolts of devices and gadgets. Technology is a culture-making activity in which human beings exercise freedom and responsibility. Besides their intended usage, technological artifacts have a bias and shape the world in economic, social, aesthetic, and cultural ways, to call only a few.

The media theorist Marshall McLuhan summarized this in his famous aphorism “The medium is the message.” This phrase suggests that media carry more than just content, they bring with them new ways of thinking, working, and communicating. John Culkin summarized McLuhan's saying with the phrase “we shape our tools, and thereafter our tools shape us.” As it turns out, our tools not only shape us, but they also shape worship and the church.

More recently, the data projector has enabled images, videos and text to be prominently displayed within the worship setting. The use of projectors to display hymns and scripture does bring advantages. Instead of looking down into their song books, people can look up at a screen and project their voices. In addition, churches can use music from many sources and new music can be introduced more easily. Once again, the projector is not only a means of displaying content, but it brings other changes. Displaying scripture on screens leads people to be less inclined to take their own Bibles and follow along. Projecting songs on a screen can reduce musical literacy since the medium tends to exclude the musical notes traditionally found in hymn books. No longer can small children follow along with their parents pointing out the words to a song.

- Derek Schuurman

Contents

| | |
|---------------------------------------|----|
| Declaration | 2 |
| Acknowledgement | 3 |
| Abstract | 4 |
| Contents | 5 |
| CHAPTER 1 | 6 |
| INTRODUCTION | 6 |
| 1.1 Introduction | 6 |
| 1.2 Objective | 6 |
| 1.3 Motivation | 6 |
| 1.4 Overview of the Project | 6 |
| 1.5. Chapter wise Summary | 7 |
| Chapter-2 | 8 |
| Analysis and Design | 8 |
| 2.1 Functional Requirement | 8 |
| 2.2 Non-Functional Requirement | 8 |
| 2.3 Architecture | 9 |
| 2.4 Use Case Diagram | 10 |
| 2.5 Sequence Diagram | 11 |
| Chapter-3 | 12 |
| Implementation | 12 |
| 3.1 Modules Description | 12 |
| 3.2 Implementation Details | 12 |
| 3.3 Tools used | 18 |
| Chapter-4 | 20 |
| Test Results/Experiments/Verification | 20 |
| 4.1 Testing | 20 |
| 4.2 Result | 24 |
| 4.3 Verification | 29 |
| Chapter -5 | 30 |
| Conclusion and Further Scope | 30 |
| 5.1 Conclusion | 30 |
| 5.2 Further Scope of Development | 30 |
| REFERENCES | 31 |

CHAPTER 1

INTRODUCTION

1.1 Introduction

SAC-Scripture And Canticle is a software created by Dazzle Info Media (currently known as Trinitech Dazzle), can be used in all the churches. This software specifically done for medium and large size churches. This can be used to give enthusiastic movements to the members in the churches, and it will improve the status of the church and the church will get the name as Computerized Church. During the song session this will give the pleasure to the members in the church.

1.2 Objective

SAC-Scripture And Canticle is an on-premises bible verse and song lyrics presentation software (in Tamil and English) which has been created 15 years back, now a days, the usability of the software has decreased dramatically because of the old design and structure. The software is basically created to use in projector which are now replaced by LED TVs. the challenge now is to upgrade the software to the current trend. To do this project I have got the legal permission from the respective company.

1.3 Motivation

SAC-Scripture And Canticle has been used in my church for almost a decade. First, this software is displayed through the projector, which is the best use case for this software, and then as technology and innovation improvise, projectors are replaced with LED TVs. in which this software faced many technical problems and design flaws, so I decided to update this software to the current trend.

1.4 Overview of the Project

In this project, I am going to move the on-premises software into a web application which is a lot more productive and then I am going to improve the design and structure to its zenith and make it up to the current trends. the presentation which are now in black text and white background will be changed into suitable texture and will be made into an easier interface. I am going to use azure app service, Azure Database for MySQL, and Visual studio to do this project.

1.5. Chapter wise Summary

Chapter 1 – Introduction

This chapter includes the introductory part such as Introduction, Objective, Motivation and Overview of the project.

Chapter 2 – Analysis and Design

This chapter includes the technical aspects like Functional Requirement, Non-Functional Requirement, Architecture, Use case Diagram and Sequence Diagram of the project.

Chapter 3 – Implementation

This chapter includes the performance part such as Modules Description, Implementation Details, Tools used in the project

Chapter 4 – Test Results/Experiments/Verification

This chapter includes the testing phase of the project such as Testing, Results and Verification.

Chapter 5 – Conclusion and Further Scope

This chapter has the conclusive part of the project which is the Conclusion and Further Scope.

Chapter-2

Analysis and Design

2.1 Functional Requirement

Requirements and Analysis is a very crucial part of any project. Without analysis, collecting data or a good planning a project will never be completed. A project must be planned before the execution or the implementation for better efficiency and accuracy.

Functional Requirement is the foundation of the project which describes what the product must do, and it is a mandatory requirement should be mentioned in a project. The main objective of this, is to help you verify the functionality of the software, which is nothing but the feature of the product and to capture the use case, which is illustrated in the following use case diagram.

Here, the Functional Requirement is that the product must produce the bible verse and songs, which are stored in the Database, in a front-end manner. Functional requirements define the internal workings of the software: that is, the technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied. They are supported by non-functional requirements, which impose constraints on the design or implementation.

2.2 Non-Functional Requirement

Non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that specify specific behaviour or functions. Typical non-functional requirements are reliability, scalability, and cost. Non-functional requirements are often called the utilities of a system.

Non-functional requirements, also known as quality attributes, describe the general software characteristics, it is a non-mandatory requirement which can be or not in the project. The main objective of this is to help you verify the performance of the software, which is product properties and to know as a quality attribute.

A reliable model would be able to catch any errors and exception and should prevent the system, from crashing. The model developed here can do so. The addition of various functionalities and modules in the future is possible therefore making it scalable. The model is developed using various free to use software packages which means that the development of the model is cost effective.

2.3 Architecture

Sometimes, having two major tiers is not enough for the system to function well. Therefore, there is a need to have three tiers, one of them being the business side. Practically, the three-tier architecture consists of three layers, including the presentation layer, business layer, and data layer. These layers are independent and might run on different servers. Moreover, each layer is treated as a separate module, including independent development and maintenance. Below, each tier is going to be discussed in more detail.

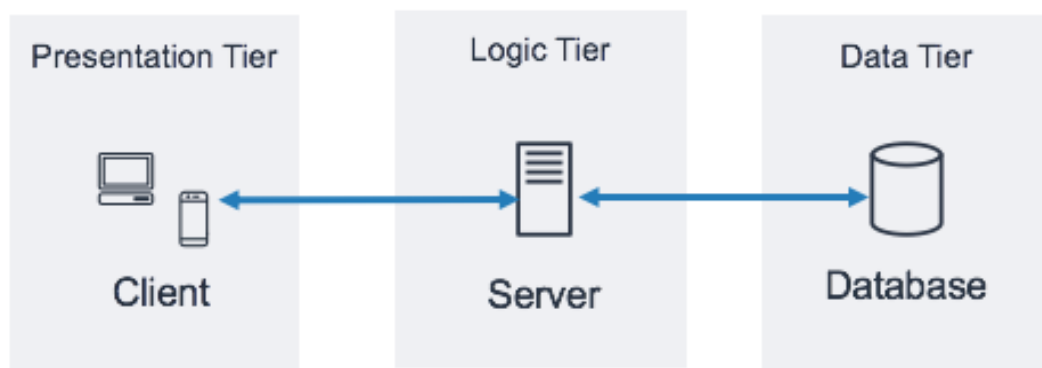


Figure 2.1

2.4 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent: Scenarios in which your system or application interacts with people, organizations, or external systems; Goals that your system or application helps those entities (known as actors) achieve the scope of your system.

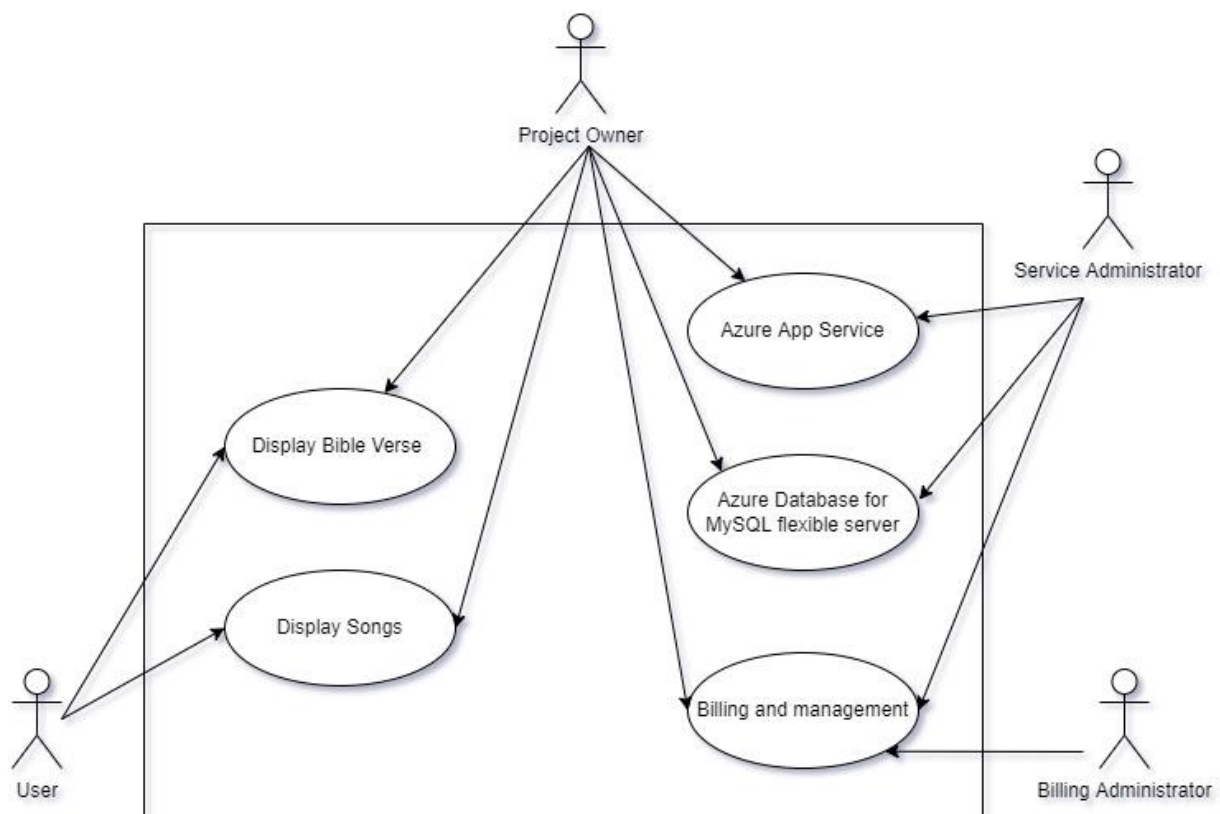


Figure 2.2

2.5 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

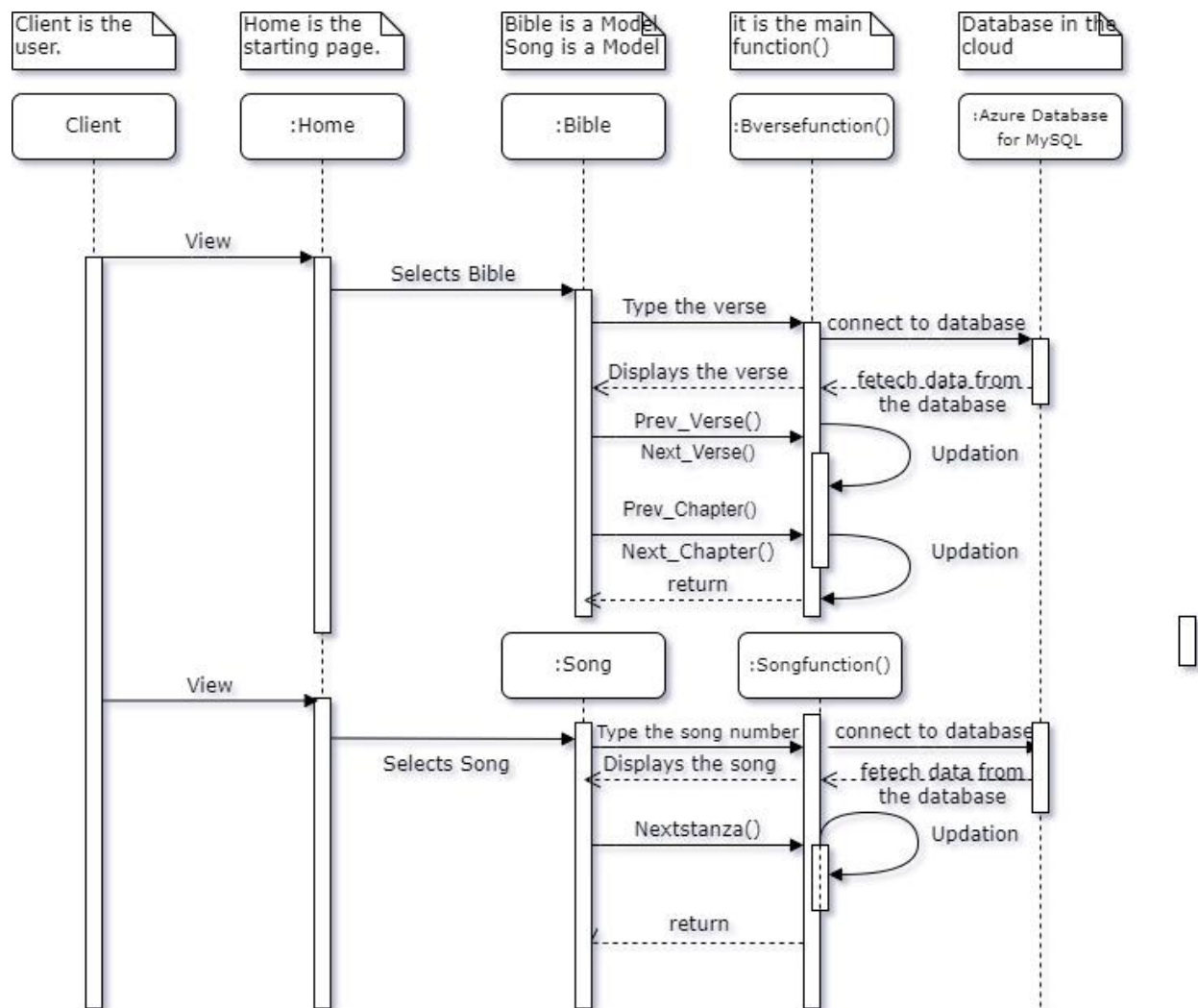


Figure 2.3

Chapter-3 Implementation

3.1 Modules Description

The first and foremost module is the SQL Server Setting module basically which connects to the MySQL Database. This module gets the hostname, username, and password from the Web.config file and it will connect to the server accordingly. This is the main objective of this module.

There are four model in this project which are Tamil bible, English bible, Tamil song and English song in which both English and Tamil bible are close to each other but have access to different databases and English and Tamil song is same as the bible model.

The Bible model has six modules which are Bversefunction(), Get_Verse(), Next_Verse(), Prev_Verse() , Next_Chapter(), Prev_Chapter(). The Bversefunction() is the main function which fetches the bible verse for the database and other function given to their name executes their function. Next_Verse() is to go to the next verse. Prev_Verse() is to go to the previous verse. Next_Chapter() is to go to the next chapter and Prev_Chapter() is to go to the previous chapter.

The Song model has two modules which are Songfunction() and Nextstanza(), here the main function is the Songfunction() which retrieves the song from the database. Coming to another one, from the name itself we can tell that its job is to go to the next stanza of the song.

3.2 Implementation Details

In this module, I am going to include the implementation details of the above mention modules.

3.2.1 SQL server settings

```
using System;
using System.Configuration;

namespace Sac_3._0_prototype.Models
{
    public class SQLServerSettings
    {
        public SQLServerSettings()
        { }
        public static String GetConnectionString(String dbName)
```

```

        {
            String serverName = ConfigurationManager.AppSettings["host"];
            String userName = ConfigurationManager.AppSettings["userid"];
            String password = ConfigurationManager.AppSettings["password"];
            String connectionString = "server =" + serverName +
                "; Uid=" + userName + "; password =" + password + " ";
            persistsecurityinfo = True; database =" + dbName + "; SslMode = none";
            return connectionString;
        }

    }

}

```

3.2.2 Bible Model

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;

namespace Sac_3._0_prototype.Models
{
    public partial class tamilbible : System.Web.UI.Page
    {
        readonly String dbName = "tamilbible";
        readonly String titleTable = "books";
        readonly String contentTable = "bible";
        static string cname;
        static int cnumber=0,vnumber=0;

        protected void Page_Load(object sender, EventArgs e)
        {
            SetDefaultButton(TextBox, Button1);
        }

        private void SetDefaultButton(TextBox txt, Button defaultButton)
        {
            txt.Attributes.Add("onkeydown", "funfordefaultenterkey1(" +
                defaultButton.ClientID + ",event)");
        }

        protected void Bversefunction(string[] arr)
        {
            try
            {
                cname = arr[0];
                MySqlConnection con;
                MySqlCommand bookname = null;
                MySqlCommand currentbookid = null;
                MySqlCommand currentchapter = null;
                MySqlCommand currentverse = null;
                string result = null;
            }
        }
    }
}

```

```

        MySqlCommand bibleverse = null;
        string result1 = null;
        int bnumber = 0;
        try
        {
            con = new
SqlConnection(SQLServerSettings.GetConnectionString(dbName));
        }
        catch (Exception ex)
        {
            Response.Write("<script>alert(SQL Connection Failed : '" + ex.Message
+ "'</script>");
            return;
        }
        try
        {
            con.Open();
            string bookidquery = "Select bookid from " + titleTable + " where
bookshortname='" + cname + "'";
            currentbookid = new MySqlCommand(bookidquery, con);
            Object bookidobject = currentbookid.ExecuteScalar();
            if (bookidobject != null)
            {
                bnumber = Convert.ToInt32(bookidobject.ToString());
            }
            else
            {
                throw (new Exception("Book not found : " + cname));
            }
            string chapterquery = "Select chapternumber from " + contentTable + "
where bookid=" + bnumber + " and chapternumber=" + arr[1] + " and versenumber=1";
            currentchapter = new MySqlCommand(chapterquery, con);
            Object chapterobject = currentchapter.ExecuteScalar();
            if (chapterobject != null)
            {
                cnumber = int.Parse(arr[1]);
            }
            else
            {
                throw (new Exception("Chapter not found : " + arr[1]));
            }
            string versequery = "Select versenumber from " + contentTable + "
where bookid=" + bnumber + " and chapternumber=" + cnumber + " and versenumber=" +
arr[2];
            currentverse = new MySqlCommand(versequery, con);
            Object verseobject = currentverse.ExecuteScalar();
            if (verseobject != null)
            {
                vnumber = int.Parse(arr[2]);
            }
            else
            {
                throw (new Exception("Verse not found : " + arr[2]));
            }
            string booknamequery = "Select bookname from " + titleTable + " where
bookid=" + bnumber;
            bookname = new MySqlCommand(booknamequery, con);
            object booknameobject = bookname.ExecuteScalar();
            string versecontentquery = "Select verse from " + contentTable + "
where bookid=" + bnumber + " and chapternumber=" + cnumber + " and versenumber=" +
vnumber;
            bibleverse = new MySqlCommand(versecontentquery, con);

```

```

        object bibleverseobject = bibleverse.ExecuteScalar();
        result = booknameobject.ToString();
        result1 = bibleverseobject.ToString();
        header.Text = result + ":" + cnumber + ":" + vnumber;
        message.Text = result1;
        con.Close();
    }
    catch (Exception ex)
    {
        Response.Write("<script>alert('" + ex.Message + "')</script>");
        con.Close();
    }
}
catch (Exception ex)
{
    Response.Write("<script>alert('" + ex.Message + "')</script>");
}
}

protected void Get_Verse(object sender, EventArgs e)
{
    string bverse = TextBox.Text;
    string[] textSplit = bverse.Split('.');
    Bversefunction(textSplit);
}
protected void Next_Verse(object sender, EventArgs e)
{
    vnumber += 1;
    string[] textSplit = { cname, cnumber.ToString(), vnumber.ToString() };
    Bversefunction(textSplit);
}
protected void Prev_Verse(object sender, EventArgs e)
{
    try
    {
        vnumber -= 1;
        if (vnumber <= 0)
        {
            vnumber = 1;
            throw (new Exception("Verse not found : "));
        }
        string[] textSplit = { cname, cnumber.ToString(), vnumber.ToString() };
        Bversefunction(textSplit);
    }
    catch (Exception ex)
    {
        Response.Write("<script>alert('" + ex.Message + "')</script>");
    }
}
protected void Next_Chapter(object sender, EventArgs e)
{
    cnumber += 1;
    string[] textSplit = { cname, cnumber.ToString(), vnumber.ToString() };
    Bversefunction(textSplit);
}
protected void Prev_Chapter(object sender, EventArgs e)
{
    try
    {
        cnumber -= 1;
        if (cnumber <= 0)
        {

```

```

        cnumber = 1;
        throw (new Exception("Chapter not found : "));
    }
    string[] textSplit = { cname, cnumber.ToString(), vnumber.ToString() };
    Bversefunction(textSplit);
}
catch (Exception ex)
{
    Response.Write("<script>alert('" + ex.Message + "')</script>");
}
}
}
}

```

3.2.3 Song Model

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using MySql.Data.MySqlClient;
using System.Configuration;

namespace Sac_3._0_prototype.Models
{
    public partial class tamilsong : System.Web.UI.Page
    {
        readonly String dbName = "tamilsong";
        readonly String titleTable = "songname";
        readonly String contentTable = "song";
        static int songNumber, stanzaNumber;

        protected void Page_Load(object sender, EventArgs e)
        {
            SetDefaultButton(TextBox, Button1);
        }

        private void SetDefaultButton(TextBox txt, Button defaultButton)
        {
            txt.Attributes.Add("onkeydown", "funfordefautenterkey1(" +
            defaultButton.ClientID + ",event)");
        }
        protected void Songfunction(string snum, string stnum)
        {
            MySqlConnection con;
            MySqlCommand songname, songcontent, songcheck, stanzacheck;
            string titlename;
            string content;
            try
            {
                con = new MySqlConnection(SQLServerSettings.GetConnectionString(dbName));
            }
            catch (Exception ex)
            {
            }
        }
    }
}

```



```

        Response.Write("<script>alert(SQL Connection Failed : '" + ex.Message +
"'</script>");
        return;
    }

    try
    {
        con.Open();
        string songcheckquery = "Select songnumber from " + titleTable + " where
songnumber=" + snum;
        songcheck = new MySqlCommand(songcheckquery, con);
        object songcheckobject = songcheck.ExecuteScalar();
        if (songcheckobject != null)
        {
            songNumber = int.Parse(snum);
        }
        else
        {
            throw new Exception("Song not found : ");
        }
        string stanzacheckquery = "Select stanzanumber from " + contentTable + "
where songnumber=" + songNumber + " and stanzanumber=" + stnum;
        stanzacheck = new MySqlCommand(stanzacheckquery, con);
        object stanzacheckobject = stanzacheck.ExecuteScalar();
        if (stanzacheckobject != null)
        {
            stanzaNumber = int.Parse(stnum);
        }
        else
        {
            throw new Exception("Stanza not found : ");
        }
        string songnamequery = "Select songname from " + titleTable + " where
songnumber=" + songNumber;
        songname = new MySqlCommand(songnamequery, con);
        object songnameobject = songname.ExecuteScalar();
        titlename = songnameobject.ToString();
        string songcontentquery = "Select para from " + contentTable + " where
songnumber=" + songNumber + " and stanzanumber=" + stanzaNumber;
        songcontent = new MySqlCommand(songcontentquery, con);
        object songcontentobject = songcontent.ExecuteScalar();
        content = songcontentobject.ToString();
        header.Text = titlename;
        var outputHtml = content.Replace("\r\n", "<br />").Replace("\n", "<br
/>").Replace("\r", "<br />");
        message.Text = outputHtml;
    }
    catch (Exception ex)
    {
        Response.Write("<script>alert('" + ex.Message + "'</script>");
        con.Close();
    }
}

protected void Nextstanza(int stznumber)
{
    stanzaNumber = stznumber;
    Songfunction(songNumber.ToString(), stanzaNumber.ToString());
}

```

```

protected void Get_Song(object sender, EventArgs e)
{
    string songnum = TextBox.Text;
    string stanzanum = "1";
    Songfunction(songnum, stanzanum);
}
protected void ButtonIN_Click1(object sender, EventArgs e)
{
    Nextstanza(1);
}
protected void ButtonIN_Click2(object sender, EventArgs e)
{
    Nextstanza(2);
}
protected void ButtonIN_Click3(object sender, EventArgs e)
{
    Nextstanza(3);
}
}
}

```

3.3 Tools used

I have created this project using ASP .NET MVC for framework and MySQL Workbench for the database management and I uses the Microsoft Azure App Service, Azure Database for MySQL, and Visual Studio to implement this project. I have used HTML, CSS, JAVASCRIPT for the front-end and C# for the back end.

ASP.NET MVC: ASP.NET MVC is an open-source software from Microsoft. Its web development framework combines the features of MVC (Model-View-Controller) architecture, the most up-to-date ideas and techniques from Agile development and the best parts of the existing ASP.NET platform.

MySQL Workbench: MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modelling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more.

Azure App Service: Azure App Service enables you to build and host web apps, mobile back ends, and RESTful APIs in the programming language of your choice without managing infrastructure. It offers auto scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo. Learn how to use Azure App Service with our quick starts, tutorials, and samples.

Azure Database for MySQL: Azure Database for MySQL is a relational database service powered by the MySQL community edition. You can use either Single Server or Flexible Server to host a MySQL database in Azure. It's a fully managed database as a service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.

Visual Studio: Visual Studio is an integrated development environment (IDE) is a feature-rich program that supports many aspects of software development. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to enhance the software development process.

Chapter-4

Test Results/Experiments/Verification

4.1 Testing

Software testing is the act of examining the artifacts and the behaviour of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. In this testing phrase, I have included the general view of the project.

4.1.1 Index:

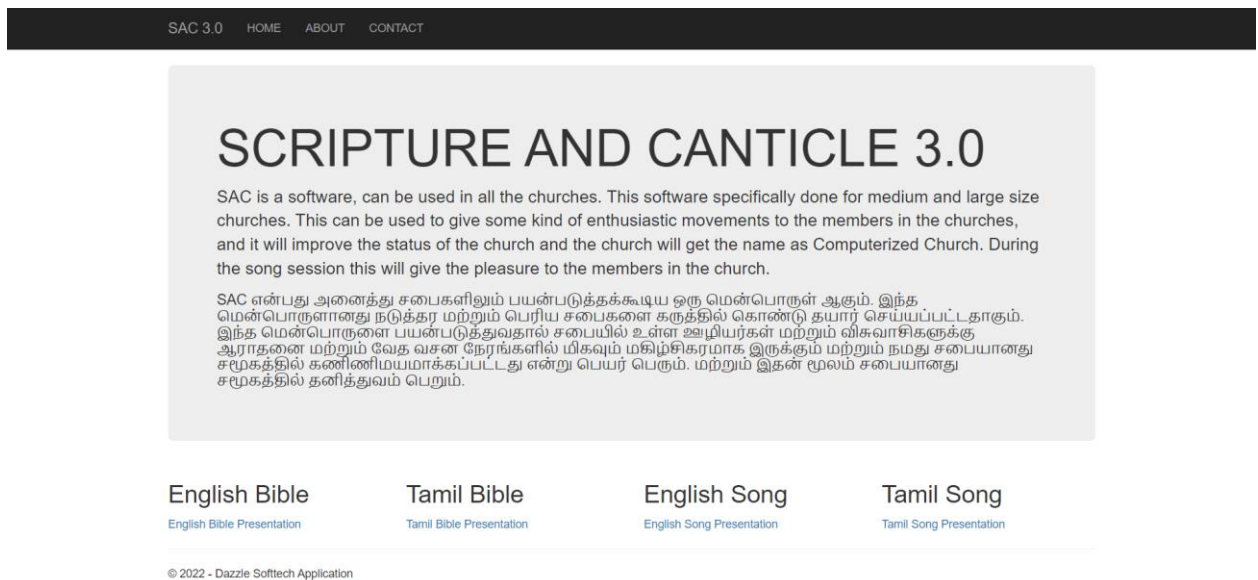


Figure 4.1

4.1.2 About

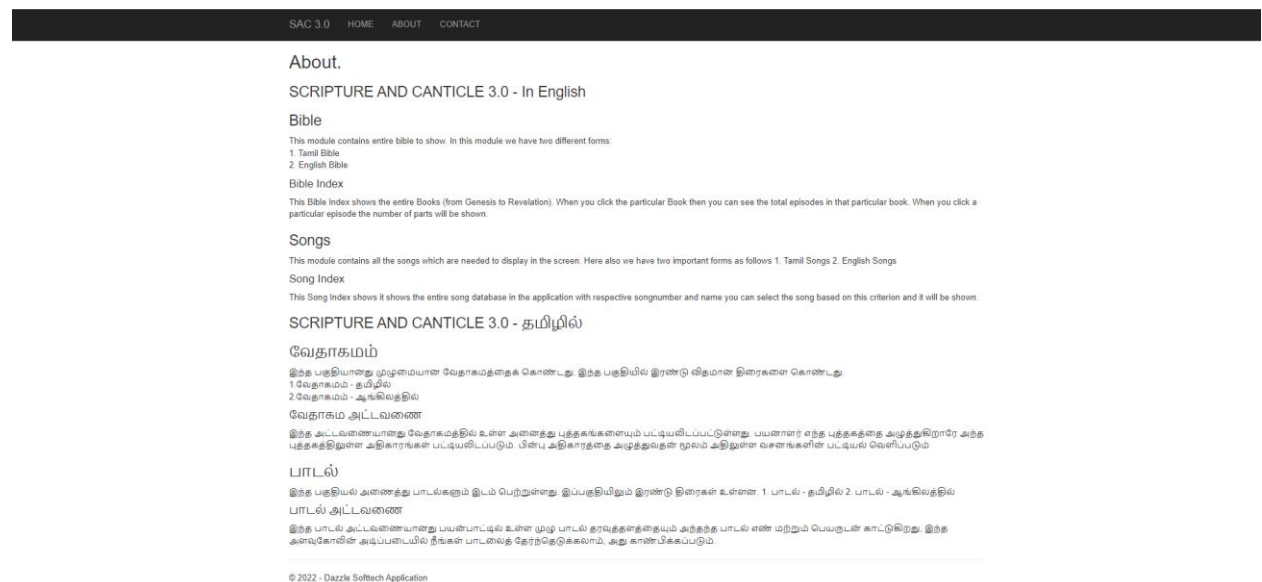


Figure 4.2

4.1.3 Contact

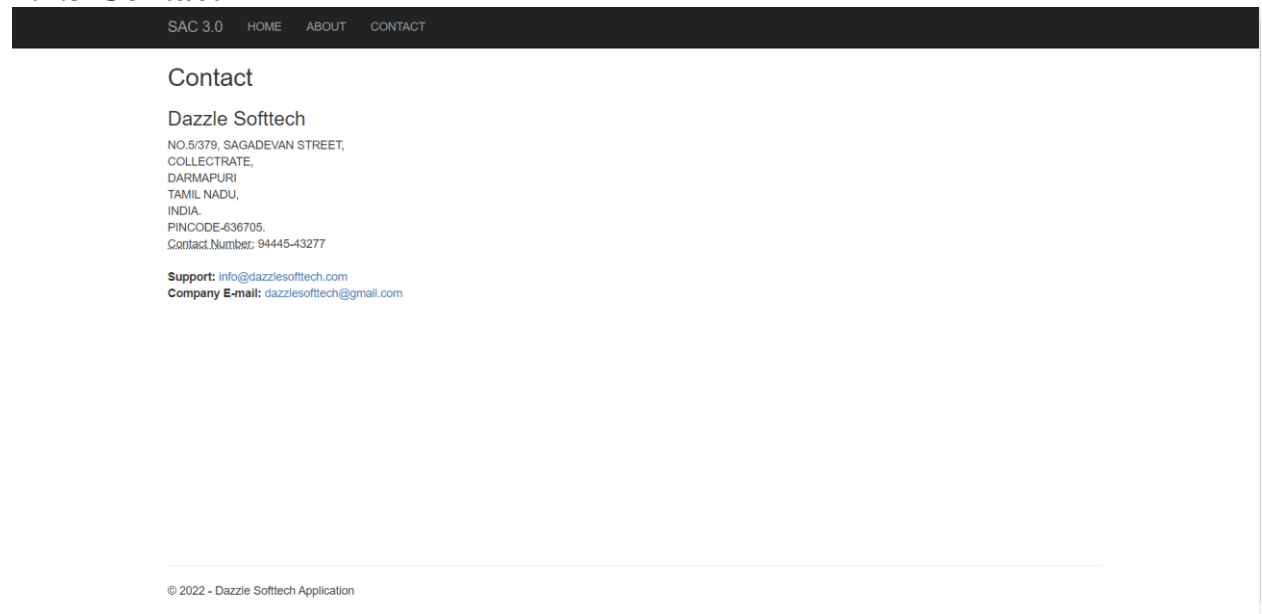


Figure 4.3

4.1.4 Bible Verse (Both in English and Tamil)

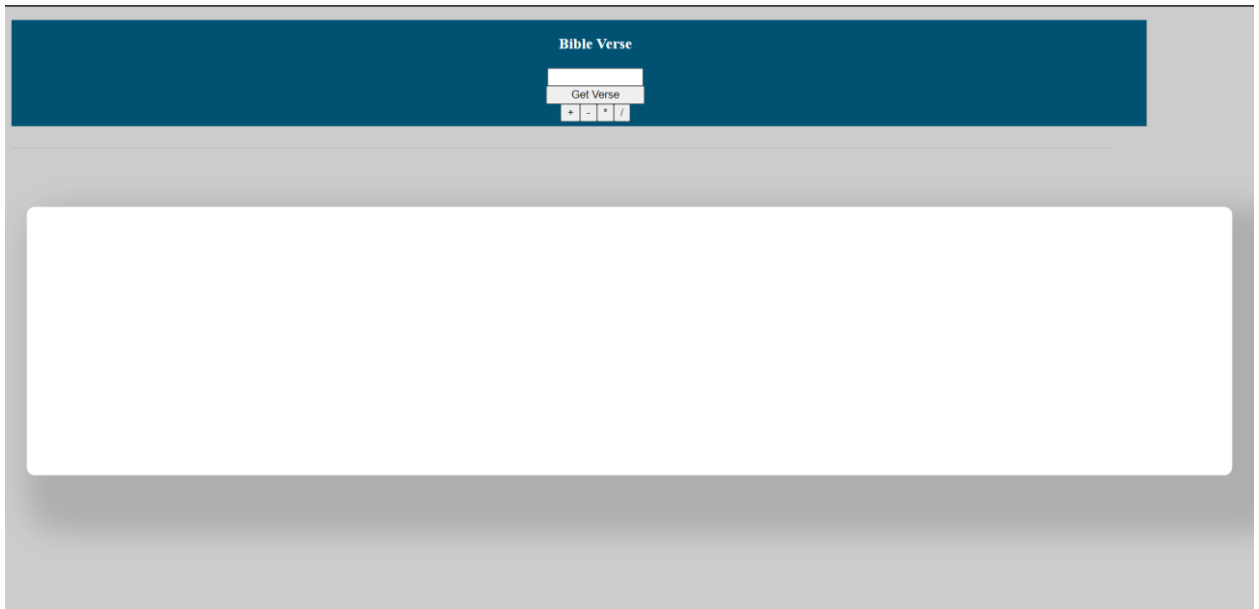


Figure 4.4a

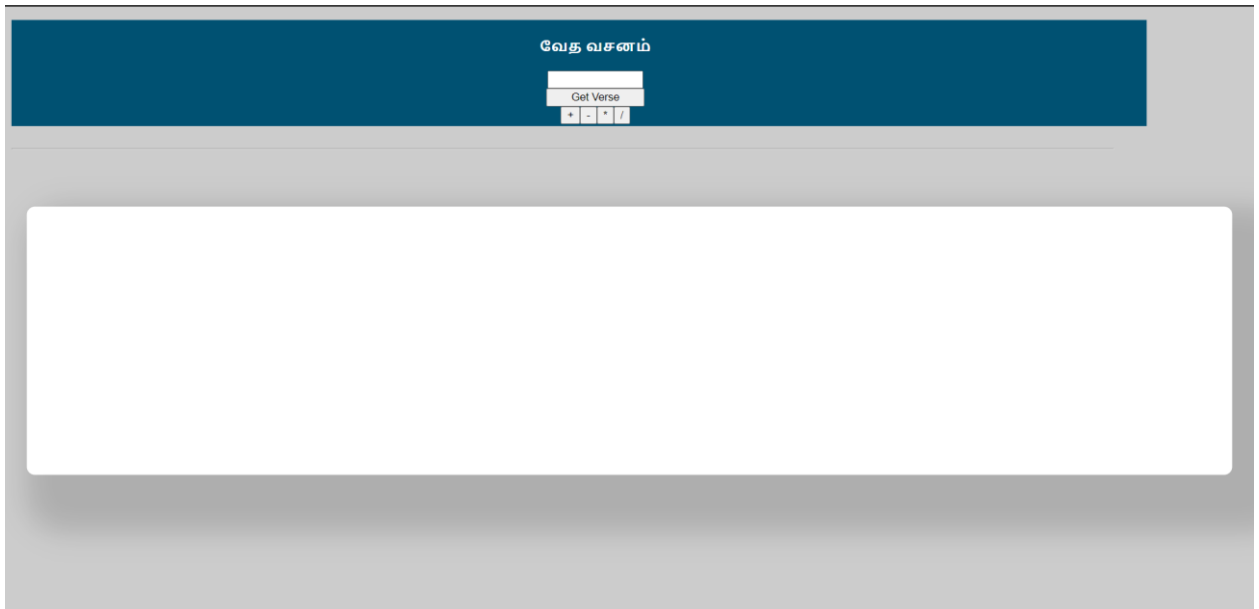


Figure 4.4b

4.1.5 Song (Both in English and Tamil)

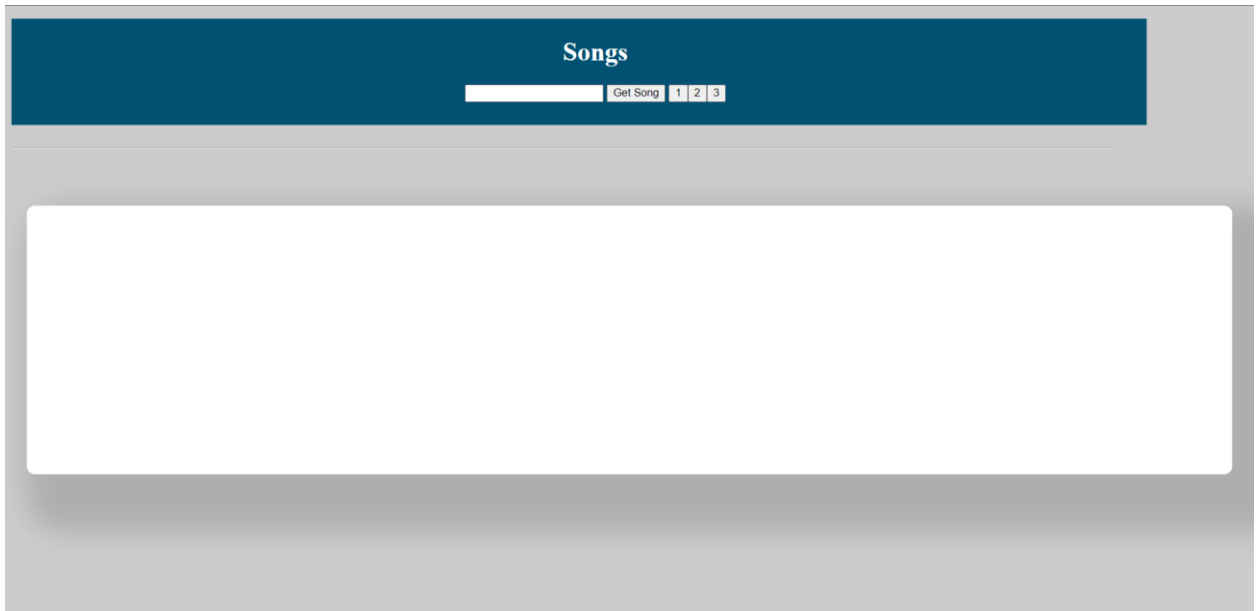


Figure 4.5a

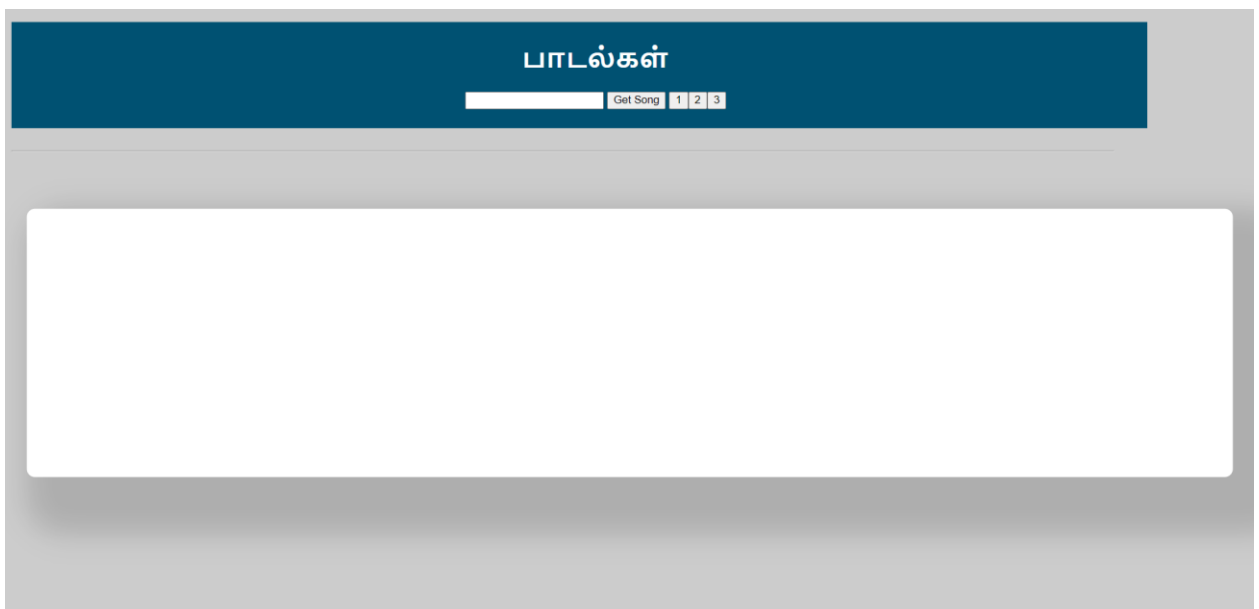


Figure 4.5b

4.2 Result

4.2.1 Bible Verse

1. The main objective of the Bible Verse Model is to display the bible verse for the given input. Here is an example, I need the bible verse from Deuteronomy chapter 1, verse 1, I need to the input as **deu.1.1** and the verse from that book and chapter must be displayed.

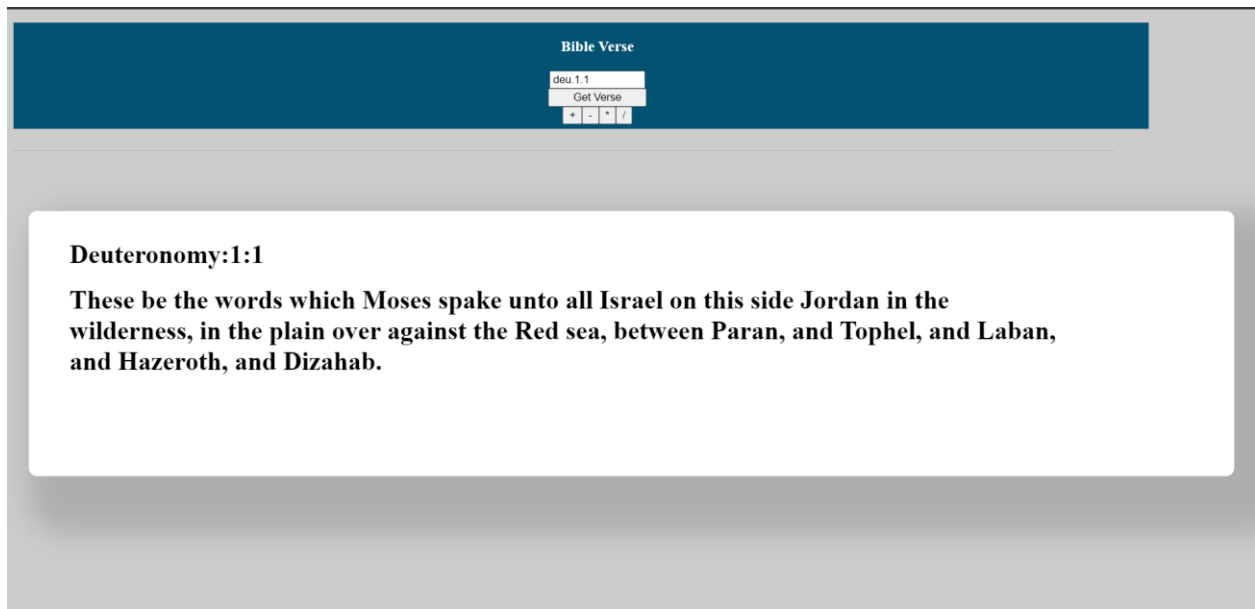


Figure 4.6a

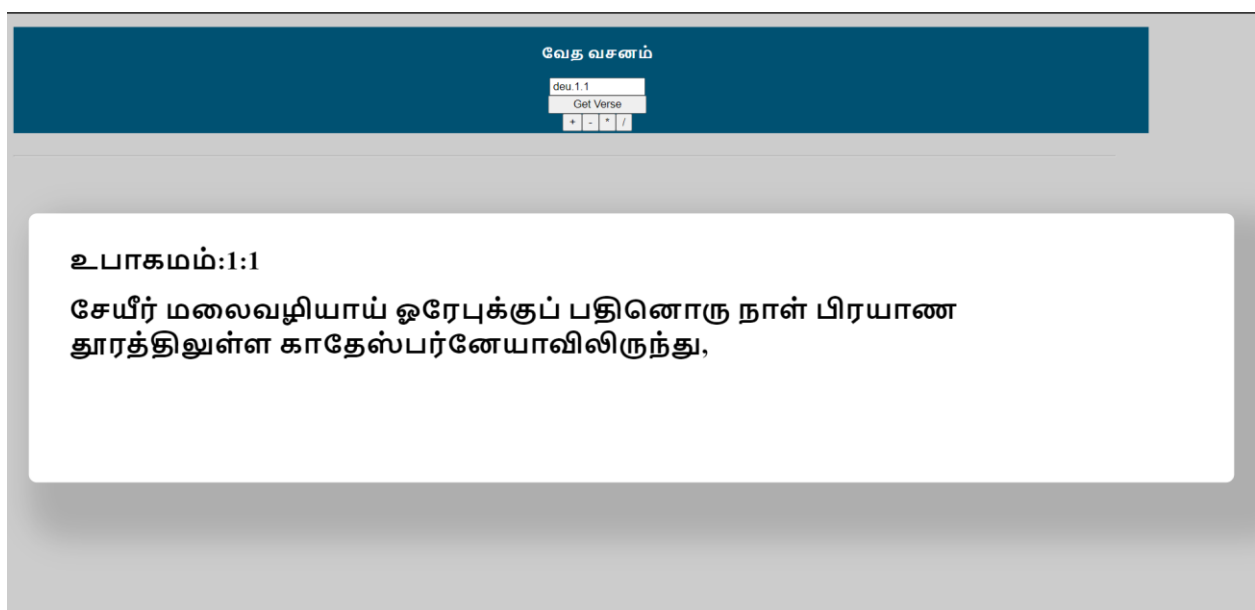


Figure 4.6b

2. There are four buttons in the top bar, of which “+” button is used to go to next verse and “-” button is used to go to the previous verse.

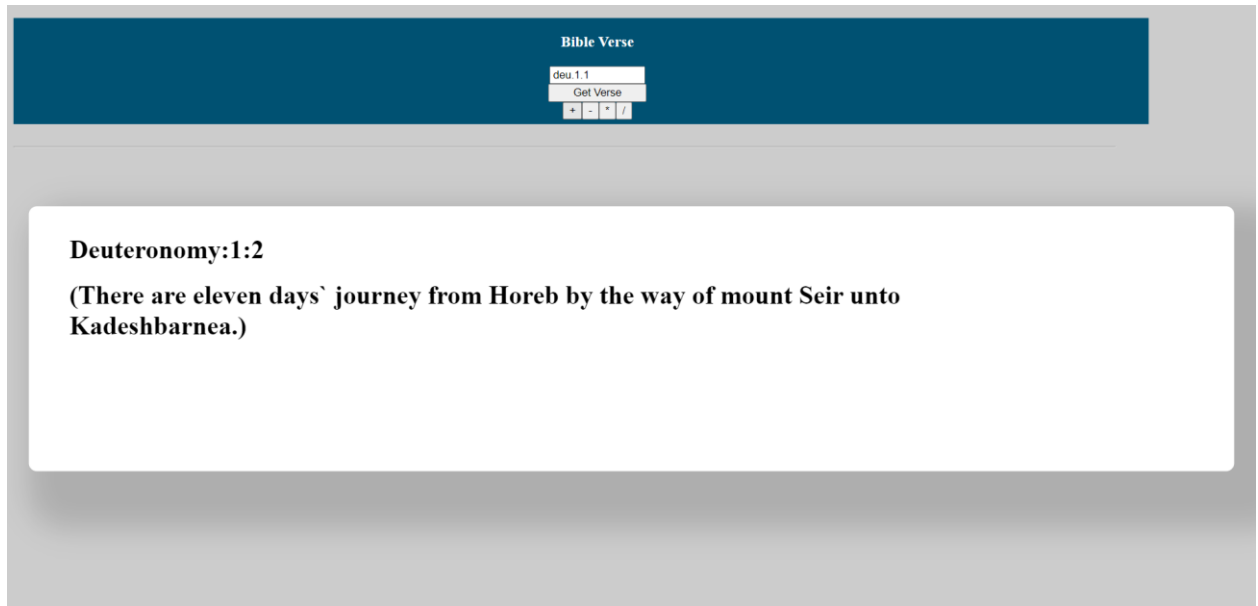


Figure 4.7a - “+” button is used.

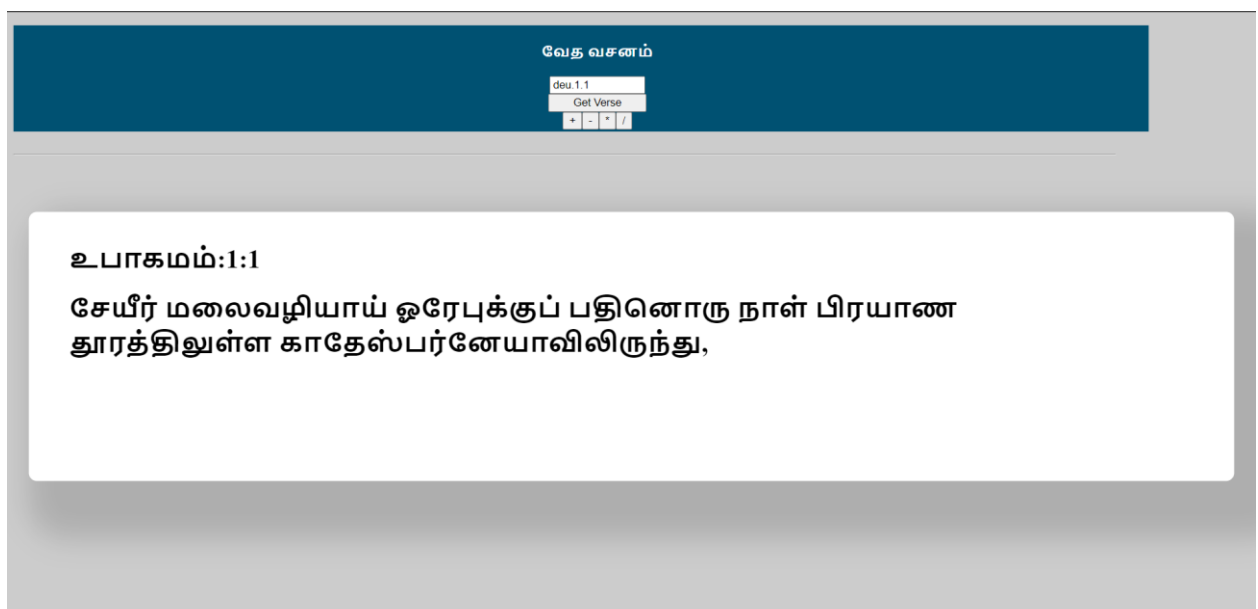


Figure 4.7b – “-” button is used.

3. The other two buttons are “*” and “/” which are used to go to next chapter and previous chapter.

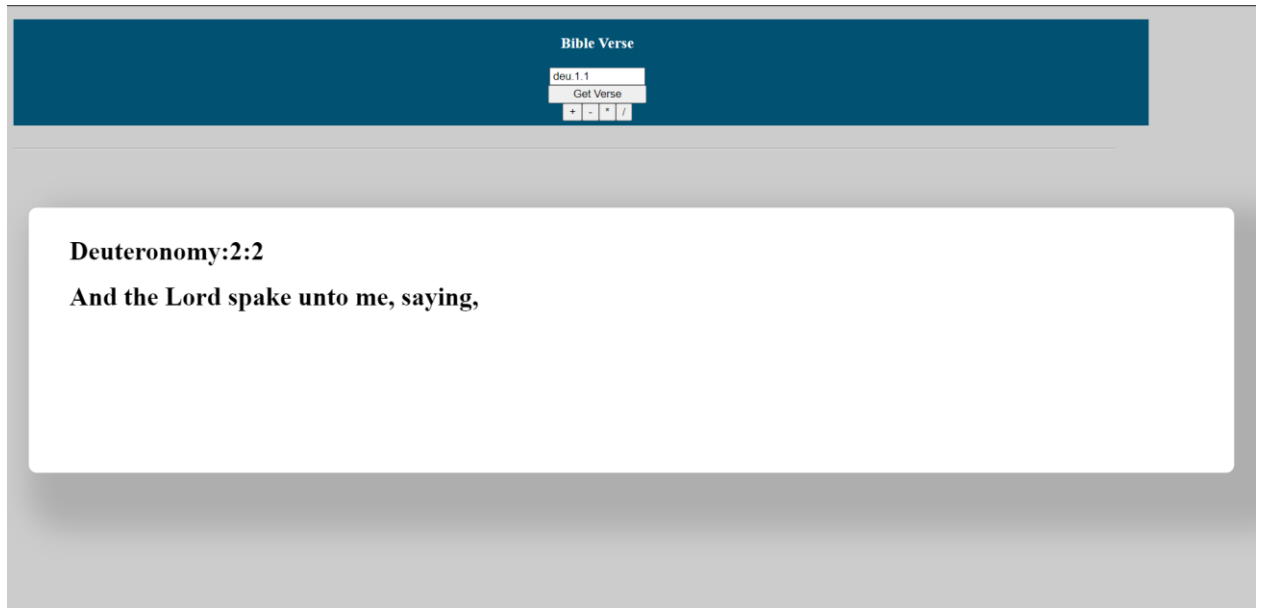


Figure 4.8a – “*” button is used.

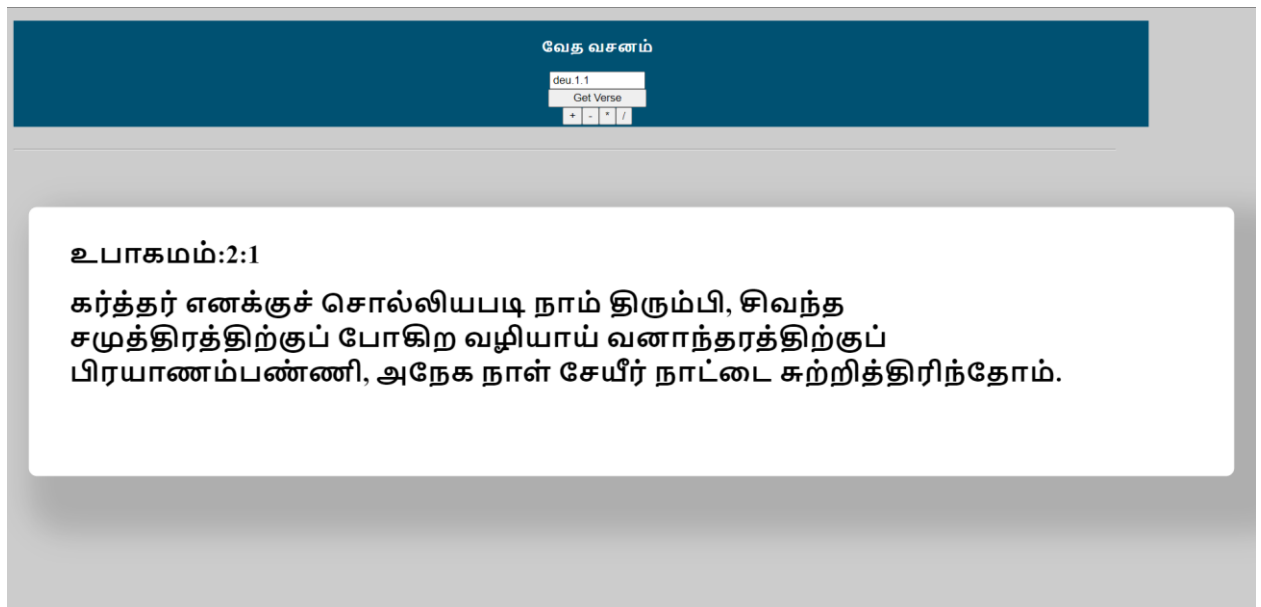


Figure 4.8b – “/” button is used.

4.2.2 Songs

1. The main objective of the Song Model is to display the song for the given input. Here is an example, I need to display the 1st song in my database, so, I need to the input as **1** and the song from the database must be displayed.

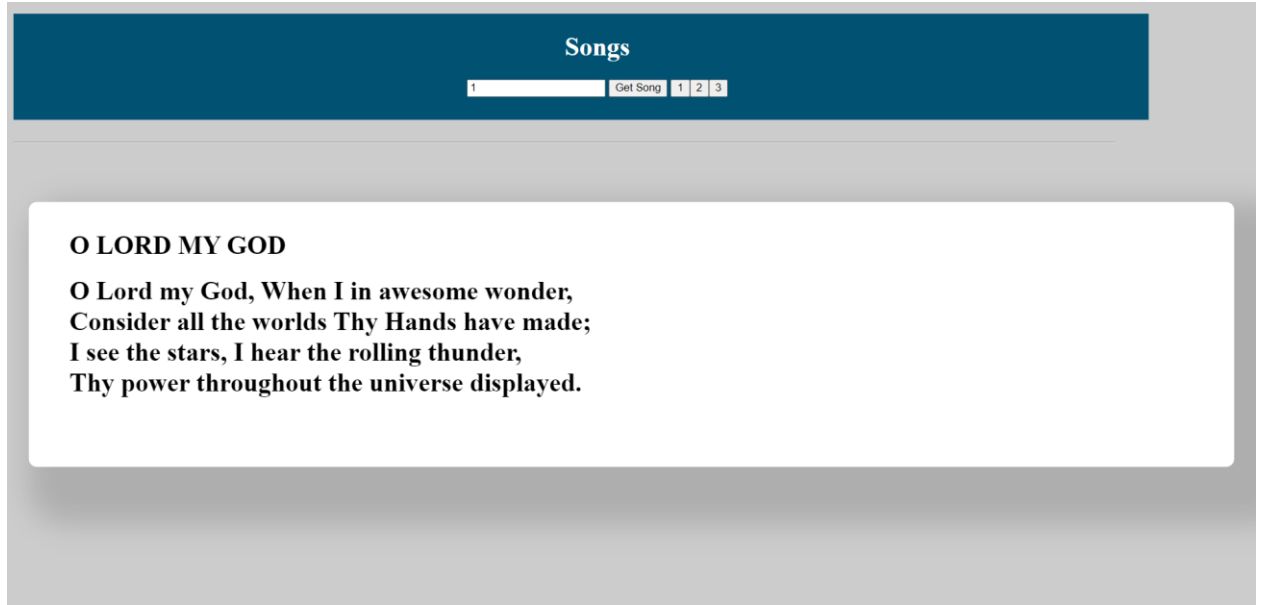


Figure 4.9a

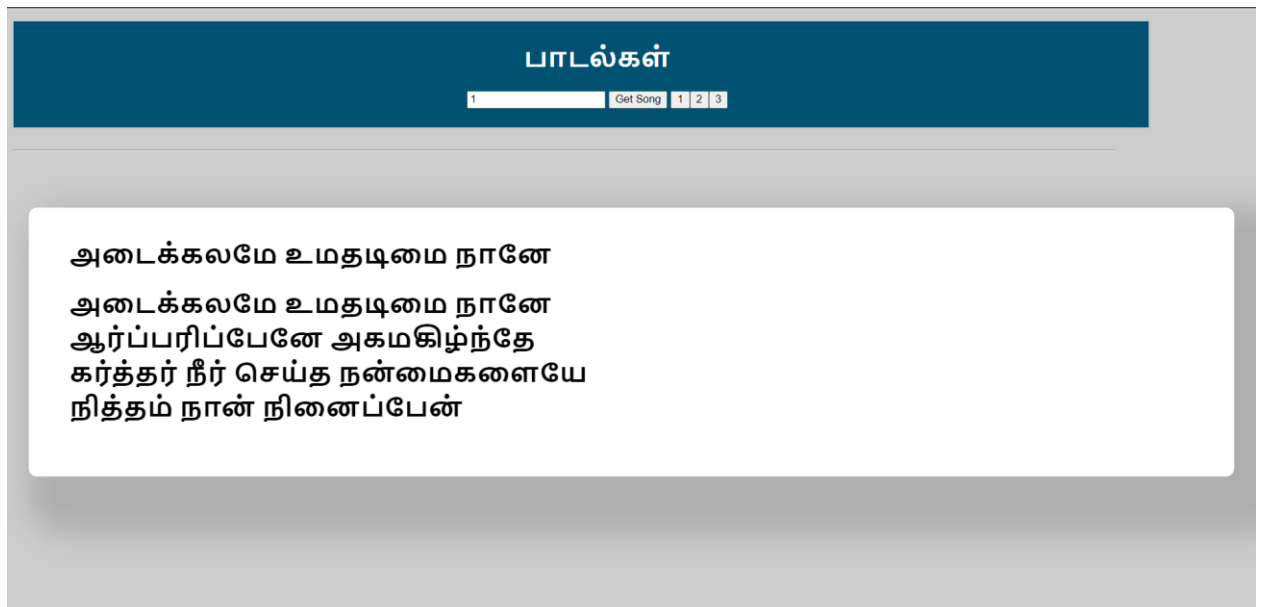


Figure 4.9b

2. There are three buttons in the top bar which is used to move forth and to the stanza of the song.

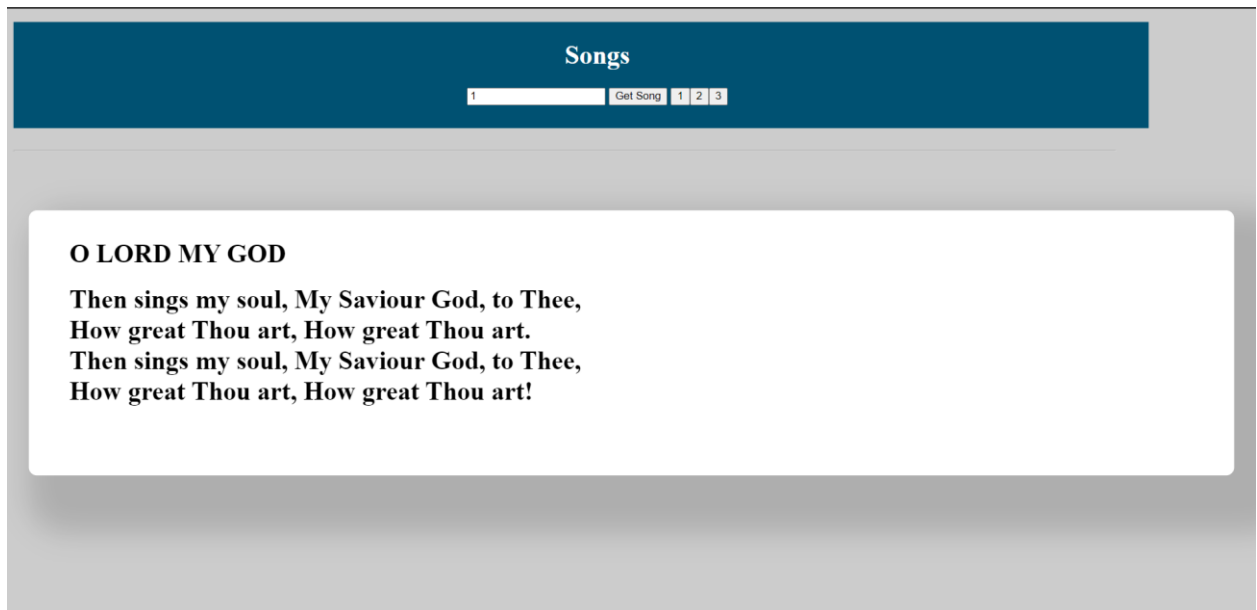


Figure 4.10a – From 1st stanza to 2nd stanza

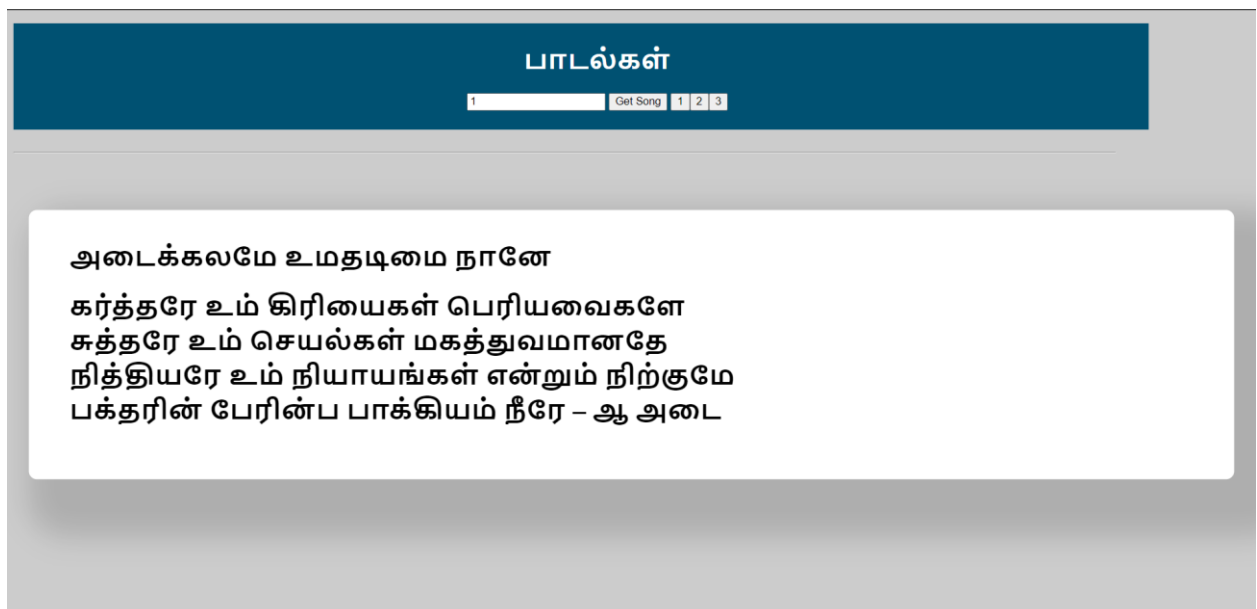


Figure 4.10b – From 1st stanza to 3rd stanza

4.3 Verification

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

| Test Case | Test Case Description | Test Data | Expected Result | Actual Result | Pass/Fail |
|-----------|-------------------------------|-----------|--|--|-----------|
| 1 | Bible Verse- Correct Format | Ge.1.1 | Bible verse from Genesis chapter 1, verse 1. | Bible verse from Genesis chapter 1, verse 1. | Pass |
| 2 | Bible Verse – Invalid Book | Chris.1.1 | Book not found : Chris | Book not found : Chris | Pass |
| 3 | Bible Verse – Invalid Chapter | Ge.70.1 | Chapter not found : 70 | Chapter not found : 70 | Pass |
| 4 | Bible Verse – Invalid Verse | Ps.128.7 | Verse not found : 7 | Verse not found : 7 | Pass |
| 5 | Bible Verse – null format | Null | Book not found : | Book not found : | Pass |
| 6 | Song – Correct format | 1 | 1 st Song in the database | 1 st Song in the database | Pass |
| 7 | Song – Invalid song number | 3 | Song not found : | Song not found : | Pass |

Chapter -5

Conclusion and Further Scope

5.1 Conclusion

Having developed this project (SAC-3.0), I have learned so many things, one such is what is MVC(Model-View-Controller) and how to develop a product using agile development methods and how to integrate and implement this in the Cloud using Microsoft Azure.

5.2 Further Scope of Development

I have merely only adopted the basic modules of the software and I have yet to adopt the remaining modules which include Adding and Searching of bible verse and songs, Bible and Song Profile's Creation, Edit and Update from the root version. And since it is a Web-Application, I do like to add a new feature, User Authentication so that many users can have their own version of this software and Separate Database.

REFERENCES

- [1] A. Berson and S. Smirth , *Transients in Power Systems*, 3rd Edition, McGraw Hill, 1998, ISBN:81-265-0280-0.
- [2] Larry M.Stephens, "Consensus Ontologies in Web Page Design", *IEEE Transactions On Pattern Recognition and Machine Intelligence*, vol.29, no.4, pp. 120-135, September 1999.
- [3] K.E.Barner, "Joint Region Merging Criteria", *Proceedings of International Conference on Image Processing*, vol.25, pp. 108-111, Sept.2007.
- [4] [online] <http://www.vicomsoft.com/knowledge/reference/firewalls1.html>.
- [5] [online] <https://app.diagrams.net/>
- [6] [online] <http://www.mayimsystech.com/products.php>
- [7] [online] <https://godandnature.asa3.org/essay-Isquottechnology-and-the-church-by-derek-schuurman.html>
- [8] [GitHub Repository Link] <https://github.com/chrispaul07/Sac-3.0-prototype>
- [9] [Project Hosting Link] <https://sac30-prototype.azurewebsites.net/>