

Developing an Application as a Learning Supplement for Key Concepts in x86\_64  
Assembly Language Programming using NASM

By  
Maya Swaminathan

A thesis presented to the Honors College of Middle Tennessee State  
University in partial fulfillment of the requirements for graduation  
from the University Honors College

Spring 2023

Thesis Committee:

Dr. Suk Jai Seo, Thesis Director

Dr. Philip Phillips, Thesis Committee Chair

Developing an Application as a Learning Supplement for Key Concepts in x86\_64  
Assembly Language Programming using NASM

by Maya Swaminathan

APPROVED:

---

Dr. Suk Seo, Thesis Director  
Professor, Computer Science

---

Dr. Philip Phillips, Thesis Committee Chair  
Associate Dean, University Honors College

# **Abstract**

At Middle Tennessee State University, the Introduction to Assembly Language (CSCI 3160) course is open to students from different fields of study. If students need extra help with the content of the course outside of class, there are not many resources that they can use to effectively learn and gain additional practice. We observed that some students struggle and possibly fail the course. Although office hours are available, some students do not utilize them. Previous research shows that a supplemental application can benefit students finding difficulty with a course. For those reasons, the author has developed an application as a learning supplement for students struggling with key concepts in the course. The creation of the supplement consists of two phases: research and implementation. The resulting application is a user-friendly interface that covers thirteen topics crucial to the course's successful completion. Students can now utilize the application to get extra help for the course. We believe that the application will improve the pass rate for the course.

# Table of Contents

<b>Introduction.....</b>	<b>1</b>
<b>Specific Aims .....</b>	<b>3</b>
<b>Methodology .....</b>	<b>3</b>
<b>Aim 1.....</b>	<b>4</b>
<b>Objective 1.1.....</b>	<b>4</b>
<b>Objective 1.2.....</b>	<b>4</b>
<b>Aim 2.....</b>	<b>5</b>
<b>Objective 2.1.....</b>	<b>5</b>
<b>Objective 2.2.....</b>	<b>6</b>
<b>Objective 2.3.....</b>	<b>7</b>
<b>Results .....</b>	<b>8</b>
<b>User Interface .....</b>	<b>8</b>
<b>Major Features .....</b>	<b>14</b>
<b>Discussion.....</b>	<b>16</b>
<b>Conclusion .....</b>	<b>17</b>

# Introduction

At Middle Tennessee State University, the Introduction to Assembly Language (CSCI 3160) course is open to students from different fields of study. The students in CSCI 3160 learn to program in Assembly x86\_64, a programming language that is very close to machine language, which is the only language that a computer can understand. Upon completion of CSCI 3160, students are expected to be able to describe different ways data is represented in a computer and to code programs in x86\_64 Assembly language to solve various problems.

The course requires students to compile their programs with Netwide Assembler (NASM), which is the medium used to translate the programmer's code. NASM is a popular assembler for compiling Assembly programs because it is open-source and portable [1]. The course also contains math components to understand structures within the Assembly x86\_64 language. For example, to understand the data representation and structures within the language, students need to grasp different number systems such as converting between binary, decimal, and hexadecimal [2].

In CSCI 3160, most students are from two different majors: Computer Science and Engineering Technology. For Computer Science students, the course is an elective, whereas for Engineering Technology students, this class is a required course. Among the Computer Science students, some have programmed with Assembly x86\_64 in another course, where they used a different compiler. Therefore, the concepts might be harder to grasp. Among the Engineering Technology majors, most of the students are unfamiliar with the technologies used in the class because these technologies are mainly used within the Computer Science department and uncommon within the Engineering Technology

department. Among both groups, there are several students who have taken the prerequisites for this class several years prior, and there are several students who might not be strong in this type of math.

If students need extra help with the content outside of class, there are not many resources they can use to supplement their learning and gain extra practice. Common computer science resources like TutorialsPoint have resources on learning Assembly language, but because the compiler used in those resources and tutorials is different from the compiler used in the class, the structure of the language differs. These resources also do not explain the math behind the programming, so students look at several resources for that supplementary support. All these factors result in students doing poorly in this class and not having resources to help them when they struggle.

A supplemental software or tutorial to remedy these challenges could be beneficial to student learning. These educational applications can provide students the flexibility to study at their own pace and help with student engagement, academic retention, and academic achievement. According to Ekaterina Pechenkina et al., the students who used a gamified mobile application as a learning tool found positive correlations between the usage of the educational application and learning retention as opposed to the students who did not use the app. These improvements to learning retention continued as students continued to use the application. The students who used the app also achieved higher scores in comparison to students who did not use the application [3].

These findings extend to undergraduate students, who would be the target group for this software. According to Kadir Demir and Ercan Akpinar, mobile learning, such as

learning through applications, can promote students' academic achievement, and students appreciate mobile learning technologies as an approach that can increase their motivation in studying a certain subject [4]. When students gave their feedback after using education-themed applications as a supplement to their learning, they said that they felt that their competency in the subjects that were the focus of the application they were using improved [5], utilizing these applications for learning assistance outside of class [6].

To provide the educational benefits to the students of the CSCI 3160 course, the author developed an application with various functionalities and features that can be used as a learning supplement to the course material. [Appendix].

## **Specific Aims**

Aim 1: Conduct research for the development of an application used as a supplement in learning x86\_64 Assembly language using the NASM compiler.

Aim 2: Develop an application that helps students understand different concepts in the CSCI 3160 course.

## **Methodology**

This project consisted of two phases, each achieving one of the specific aims.

- Research Phase: Conduct research to develop the application format and learn the usage of the tools required for the application.
- Planning and Implementation Phase: Build a detailed plan to develop an application and implement the application according to the planned schedule [7].

In the remaining section, a list of objectives is presented to demonstrate how the aims of this project have been satisfied.

## **Aim 1**

Conduct research for the development of an application used as a supplement in learning x86\_64 Assembly using the NASM compiler.

### **Objective 1.1: Research ways students learn effectively.**

The purpose of the objective was to research a structure for learning that can be reinforced by the application, specifically for its tutorial section. The task involved doing academic research on how college students learn best because the application is supposed to help college students with the contents of a college-level course. For this part of the research, the author utilized scholarly and research articles located in educational journals. We used Google Scholar and the MTSU Library for the literary search for those articles. The research was utilized when creating the content structure for the course, meaning how the content will be broken down for the user.

### **Objective 1.2: Research how to implement a user-friendly interface for education-based applications.**

The purpose of the objective was to investigate the characteristics of a user-friendly interface that enables students to learn best because if the interface is disorganized, it will result in students getting distracted and hindering them from learning the content. The research was done by finding articles in Google Scholar and the MTSU Library that describe user interface creation. The author used academic research articles published in computer science and computer graphics journals for this part of the



research. The research was utilized when creating the graphical user interface, often denoted as GUI, for the application, meaning how the content was presented on the application interface.

## **Aim 2**

Develop an application that helps students understand different concepts in the CSCI 3160 course.

### **Objective 2.1: Select the concepts covered in the application.**

We examined the content covered in the application. The first step was creating a list of topics we wanted the application to cover. The creation was done using the backward design of a course, which is when the instructor starts with learning goals and expands those learning goals to specific objectives [8]. These objectives then evolved into the more concise topics that were covered in the application. The next step was to create tutorials and practice questions for those different topics. These tutorials would utilize an active learning approach to teaching, containing the lesson as well as examples, which would keep the user engaged with the content [8] [9]. These practice questions would utilize the retrieval practice method of studying, an effective study tool for students [10].

For the application we created a set of tutorials and practice problems using class materials from the current semester as well as previous semesters. The application covers five categories of course material: 1) preliminary programming, 2) debugging, 3) data and conversions, 4) machine arithmetic, and 5) bit manipulation. For all categories except for the first one, the application includes both tutorials and practice questions. The

practice questions are presented to the user based on their difficulty levels, which were easy, moderate, and difficult.

**Objective 2.2: Design the architecture and structure of the application.**

We developed the core functionalities of the application and outlined the application's user interface and user experience (often abbreviated as UI/UX) design. This extended to developing the core functionalities of the application, which defines how the user will interact with the application. According to W. Sun, et al., the process for developing a user-friendly interface would be through analyzing what the user might want, creating a GUI design, prototyping the GUI on the choice software, and modifying the GUI software to make sure the GUI is user-friendly [11]. This process was kept in mind when developing the user interface for the application.

The application's UI was first designed on pencil and paper. This gave a basic overview of how the application looked, where the buttons would be located, and fleshed out the purpose of the different aspects of the UI. Once we had a good idea about what the application looked like, we developed the application's UI flow, which describes how the user should use the app from start to end. According to A. Shi, a navigation structure for an application should determine the order of access to the information presented in the application [12]. This means that we had to standardize the order of the information for each educational topic presented in the application.

The application's UX was then designed, which is an extension of planning the UI. It included planning the UX, which includes developing the entire user interaction with the product, which in this case, would be the application. It also considered how the

user would feel after using the app, utilizing information provided by the research from the previous aim.

**Objective 2.3: Build, test, and refine the application.**

The building, testing, and refining of the application occurred concurrently because, in the programming process, these are all done simultaneously, so errors and other issues with the software can be caught and fixed quickly. The completed project is an interactive web application where the user would be able to do tutorials and practice questions and check their answers. The user just needs to have the link to the application, accessible to users of any platform (Mac or Windows).

The first step in this process was developing the UI/UX for the application. This was done by using the outline for the UI/UX design created in the previous objective and implementing those elements into the application. The tool utilized for creating the application is a popular web development platform called Bubble, which utilizes the concepts of visual programming with JavaScript and HTML for supplemental coding [13]. The user interface was divided into three page categories: home pages containing a table of contents developed for each topic covered in the application, tutorial pages presenting the tutorials for each topic, and question pages presenting the practice questions.

The next step was developing the functionality to each of the components on the webpage. This entailed adding functionality to all the buttons in the pages and having different elements on the page react to the user input. For example, in the home page and tutorial page for each section, all the buttons had some navigation purpose, meaning that if the user clicks some button in the application, they will go to another page in the

application. Likewise, in the question pages, there were buttons that would take the user to another page in the application, but there were also buttons that were used to check the user input and give feedback to the user.

The final step was testing the application on different display types and making final changes to the application. According to R. Guntupalli, testing the user interface on several devices of different sizes to account for the different experiences that the user would have when using different devices for the application [14]. Flexibility is key to creating an engaging user interface because the application should work across devices of different sizes. Testing was done by modifying the author's Google Chrome window to make sure the content of the different pages would fit the window and making changes to the application based on the appearance of the interface. Finally, the author compared how the application would look on windows across different devices such as a tablet, laptop, and computer.

## **Results**

Using the processes in the methodology section, a web application supplementing the important concepts for x86\_64 Assembly language was created. We will describe the finished product in terms of the user interface and its major features.

### **User Interface**

We will explain the different elements on the pages of the application and the reasoning behind their placement. We will also illustrate the user interface flow or the way that the user navigates through the application. Additionally, we will demonstrate

how we standardized some of the user interface elements of the application to make the application user-friendly.

When the user first gets the link to the application, they should see the screen in Figure 1. The title screen contains the title of the application, the purpose of the application, and a button that says “start.” The *start* button would give the user access to the rest of the application.

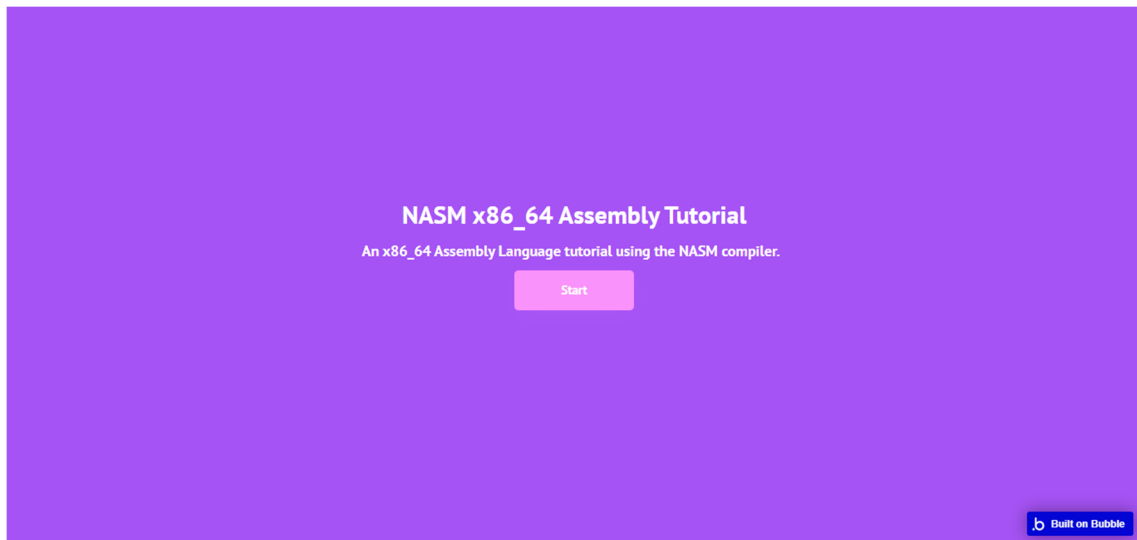


Figure 1: Application Title Page

Once the user presses the “start” button, as shown in Figure 1, the user will be directed to the Table of contents for the application, as shown in Figure 2. The table of contents contains buttons named after a topic covered in the application. A table of contents would allow the students to pick a topic they need extra help with. Each of these buttons will direct the user to the home page of the chosen topic.



Figure 2: Application Table of Contents

As stated previously, the topics covered in the application can be broken down into five categories: preliminaries, debugging, data and conversions, machine arithmetic, and bit manipulation. This categorization of the topics covered in the application is evident in the table of contents page as the topic buttons are placed under their category. This was to help the user further navigate the website. The preliminary programming category only consists of tutorial pages, whereas the other categories consist of tutorial and question pages.

As an example, if the user clicks the Number Conversions button in the application table of contents, as shown in Figure 2, they will be directed to the home page for the Number Conversions topic, as shown in Figure 3. All the home pages for the other topics look very similar, demonstrating the standardization of the home pages in the application. The home page of each topic contains an organized view of the course, which consists of sections designated for the tutorial and practice questions.

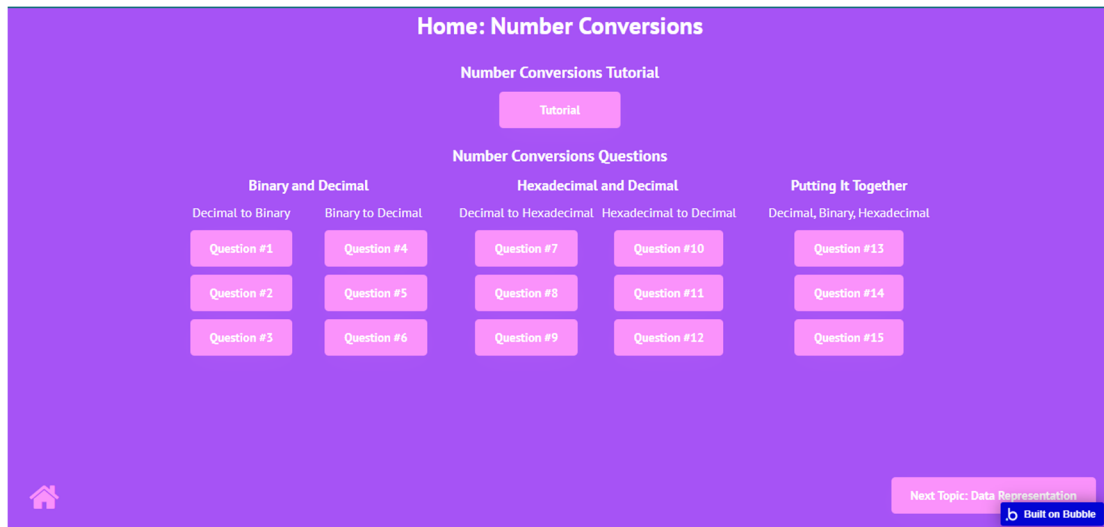


Figure 3: Number Conversions Home Page

The user has access to the tutorial button, which is positioned as the first button on the top of the page. The purpose of the tutorial pages is to allow the students to get familiar with the topic or to review the necessary concepts. The students who fully understand the concepts can skip the tutorial button and move on to work on practice questions. The application gives the user access to all the topic practice questions from the home page. In other words, it allows users to select specific questions they would like to practice without navigating through other practice questions. For example, suppose the user is having difficulty converting from decimal to hexadecimal. In that case, they can click on the buttons for questions 7, 8, and 9 and directly access the practice they would like.

At the bottom of the page, there is an icon shaped like a house. This icon is in all home pages, and if the user clicks on it, they will be taken back to the main table of contents for the application. There is also a button that says, “Next Topic: Data Representation.” This button will direct the user to the next topic in the application, in

this case, the home page for the Data Representation section. The purpose is to give the user an easy way to go to the next topic without having to go back to the table of contents for the application.

If the user clicks the tutorial button on the Number Conversions home page as shown in Figure 3, they will be directed to the tutorial for the Number Conversions topic, as shown in Figure 4. All the tutorials for the other topics covered in this application have a tutorial page that looks very similar to this. As shown in Figure 4, the tutorial page contains a PDF window that allows the user to print the tutorial PDF as well as save the PDF onto their computer for future studying purposes.

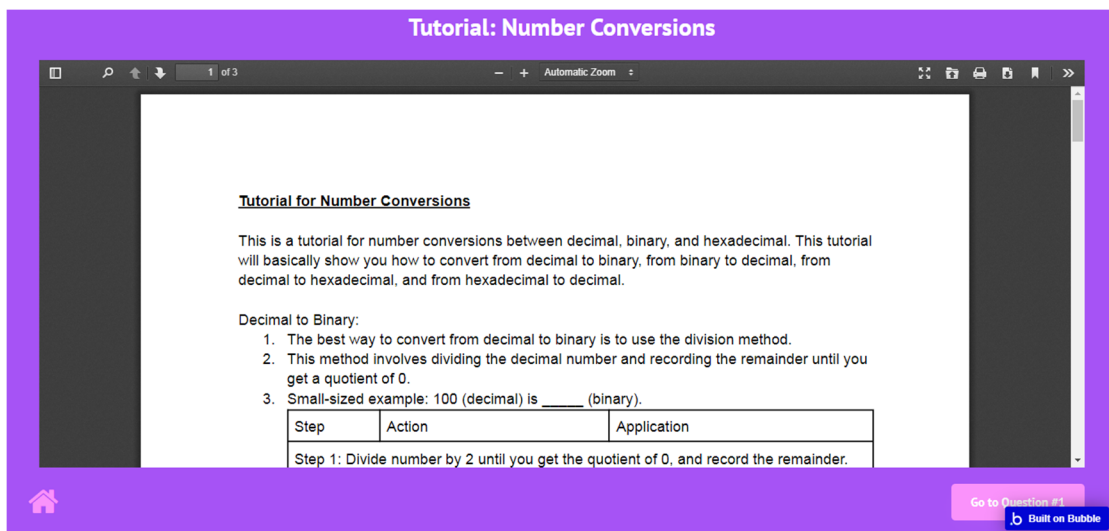


Figure 4: Number Conversions Tutorial

At the bottom of the tutorial page, as shown in Figure 4, we also see the same house-shaped icon. This icon is in all the tutorial and question pages, and if the user clicks on it, they will be taken back to the home page of the topic they have chosen. For example, if the user is on the tutorial for the Number Conversions section, they can click on the house icon, and they will be directed to the home page for the Number



Conversions section. On the bottom right we see a button that says, “Go to Question #1.”

If the user clicks that button, they will be directed to the first question in the Number Conversions section. This functionality enables the user to get to the first practice question after reading the tutorial without unnecessarily clicking many buttons.

If the user clicks on the “Question #1” button in the Questions section of the Number Conversions home page shown in Figure 3, they will be directed to the first practice question in the Number Conversions section. This same effect will be achieved if the user clicks on the “Go to Question #1” button in the Number Conversions tutorial shown in Figure 4. The page contains a series of textboxes where the user is directed to enter some input through the initial contents in each of the text boxes. The user will also see a button that says, “submit” at the bottom of the screen, which will allow the user to check their answers. This is how all the question pages will look when the user first lands on the page.

Decimal:	Binary:
3	<input type="text" value="Type here..."/>
8	<input type="text" value="Type here..."/>
22	<input type="text" value="Type here..."/>
30	<input type="text" value="Type here..."/>
52	<input type="text" value="Type here..."/>

Submit

Built on Bubble

Figure 5: Number Conversions Question #1

## Major Features

We will describe the second part of the application: major features. The major features of the application have to do with the processing of the user inputs in the practice question pages of the application. The processing of the user inputs is done through showing and hiding elements based on the user input. These major features are standardized throughout all the practice question pages in the application.

The decision to show and hide certain elements based on user input is based on a study by W. Sun, et al, where they state that implementing a user-friendly GUI includes utilizing show/hide functionality so the user will not be overwhelmed with all the information at once [11]. As stated previously, Figure 5 shows the screen that the user would encounter when they first land on a practice question. When the user clicks on the submit button, they should get a screen like the one shown in Figure 6, where each user input also has a flag icon with a color next to it. The green color indicates that the user inputted the correct answer into the textbox, whereas the red color indicated that the user inputted the incorrect answer into the textbox.

Decimal:	Binary:
3	11
8	Type here...
22	Type here...
30	Type here...
52	Type here...

Figure 6: Number Conversions Question #1 with incorrect responses

When the flags appear on the screen, the system will disable the buttons, and therefore the user cannot input anything into the textbox. In addition, at the bottom of the screen, the submit button was replaced with two buttons: “Try Again” and “Next Question.” If the user clicks the “Next Question” button, the user will be taken to the next question in the section. As demonstrated in Figure 6, the user will be directed to the second practice question in the Number Conversions section.

When the user clicks the “Try Again” button, as shown in Figure 6, the flags for the textboxes with the incorrect user input will be removed, and the user will be able to input their answers into those textboxes, as shown in Figure 7. The correct flags remain the same, so the user will not get confused about the questions they got correct and the questions they got incorrect. The application lets the user stay on the same question if they want to or until they get the answers correct.

Decimal:	Binary:
3	11
8	<input type="text"/>
22	<input type="text"/>
30	<input type="text"/>
52	<input type="text"/>

Submit

Figure 7: Number Conversions Question #1 after Try Again Button is clicked.

When the user gets all the questions correct on a page, the screen will look similar to the one in Figure 8. It will show all the flags next to the user input as correct, which is

indicated using the green-colored flag. We notice the absence of the “Try Again” button because the user got all the questions on the page correct. At this point, the user will have two options; they can either go back to the topic home page or proceed to the next question.

The screenshot shows a web application interface for a quiz titled "Question #1: Number Conversions". It features a table with two columns: "Decimal:" and "Binary:". The table contains five rows of data, each with a decimal value and its corresponding binary value. To the right of each binary value is a green flag icon, indicating that all answers are correct. At the bottom of the interface, there is a home icon, a "Next Question" button, and a "built on Bubble" logo.

Decimal:	Binary:
3	11
8	1000
22	10110
30	11110
52	110100

Figure 8: Number Conversions Question #1 with correct responses

## Discussion

With this application, students taking CSCI 3160 will have more resources to review the key concepts if they have trouble with the content or want to review the content before the exams. They can also work on practice questions on specific topics to reinforce their learning. The learning experience through the application is likely to result in students performing better in the class, but that is subject to findings when more research was done into the impact of this application on the grades of students taking the CSCI 3160 course.

Even though this application was specifically made with the CSCI 3160 in mind, the tutorials and questions could also be used in other courses such as the CSCI 3130

course, Computer Architecture and Organization. It is because the CSCI 3130 course also has an Assembly language component to it, and students who take that class often struggle with the programming because the student only gets about three weeks to learn the Assembly language required for developing their semester project.

In future research, many improvements could be made to this application. One possible improvement would be expanding the topics covered in the application by adding tutorials and questions on the jump and loop instructions in the Assembly language. Another improvement would be expanding the application's functionality so that users can save their work when using the application by creating an account on the webpage and saving the usernames and passwords in a database.

## **Conclusion**

At Middle Tennessee State University, the Introduction to Assembly Language (CSCI 3160) course is open to students from different fields of study. The course covers programming concepts for x86\_64 Assembly Language with the NASM compiler. It also contains math components to support structures within the language. Because of the diversity of the students, we often observe some students who have taken the prerequisites for this class a few years prior and others who are not strong in the math components. Unfortunately, there are not many resources available for those students to utilize to supplement their learning and gain extra practice.

An educational application can provide students the flexibility to study at their own pace and help with student engagement, academic retention, and academic achievement. For these reasons, the author developed an application as a learning supplement for students struggling with key concepts in this course. The application

utilizes research investigating how students would study and learn best in conjunction with research for developing a user-friendly interface. We then developed the application through outlining the contents of the application through creating the questions and tutorials as well as planning the user interface. We implemented the research and planning into an application utilizing the Bubble software platform.

Because of the user-friendly nature of the application, students can easily navigate through it, getting the learning supplement they need for this course. Students can now utilize this application to get extra help in the course, and they can also use the application to study for the midterm and final exams for the course. Even though this application was specifically made for CSCI 3160, the tutorials and questions could also be used in the CSCI 3130, Computer Architecture and Organization. We believe that this application will improve the pass rate for this course.

## References

- [1] NASM Team, "NASM," [Online]. Available: <https://www.nasm.us/>. [Accessed 9 October 2022].
- [2] UM Baltimore County Department of Computer Science, "The Decimal, Binary, and Hexadecimal Numbering System," 2004. [Online]. Available: [https://redirect.cs.umbc.edu/courses/undergraduate/313/fall04/burt\\_katz/lectures/Lect01/numberSystems.html](https://redirect.cs.umbc.edu/courses/undergraduate/313/fall04/burt_katz/lectures/Lect01/numberSystems.html). [Accessed 9 October 2022].
- [3] E. Pechenkina, D. Laurence, G. Oats, D. Eldridge and D. Hunter, "Using a gamified mobile app to increase student engagement, retention and academic

- achievement," *International Journal of Educational Technology in Higher Education*, pp. 14-31, 2017.
- [4] K. Akpınar and E. Demir, "The effect of mobile learning applications on students' academic achievement and attitudes toward mobile learning," *Malaysian Online Journal of Educational Technology*, pp. 48-59, 2018.
- [5] A. Camilleri and M. Camilleri, "Mobile Learning via Educational Apps: An Interpretative Study," *5th International Conference on Education and Training Technologies (ICETT)*, pp. 88-92, 2019.
- [6] D. Menon, "Uses and gratifications of educational apps: A study during COVID-19 pandemic," *Development and Educational Communication Unit (DECU), Indian Space Research Organisation*, pp. 1-12, 2022.
- [7] Codementor Team, "How to Become a Programmer: 8 Steps to Building an App from Scratch," 2015. [Online]. Available: <https://www.codementor.io/@codementorteam/how-to-build-app-from-scratch-beginner-programmer-7z0atq56w>. [Accessed 31 August 2022].
- [8] E. Dolan and J. Collins, "We must teach more effectively: here are four ways to get started," *Molecular Biology of the Cell*, pp. 2151-2155, 2015.
- [9] C. Seemiller, M. Grace, P. Dal Bo Campagnolo, I. Mara Da Rosa Alves and G. Severo De Borba, "What makes learning enjoyable? Perspectives of today's college students in the U.S. and Brazil," *Journal of Pedagogical Research*, pp. 1-17, 2021.
- [10] UC San Diego Department of Psychology, "How to Effectively Study," UC San Diego, 2022. [Online]. Available: <https://psychology.ucsd.edu/undergraduate->

program/undergraduate-resources/academic-writing-resources/effective-studying/index.html. [Accessed 14 October 2022].

- [11] W. Sun, Y. Chen, H. Boussalis, W. Liu, K. Rad and J. Dong, "On the Design and Implementation of User-Friendly Interface for Scientific and Engineering Applications," Research Gate, 2003. [Online]. Available: [https://www.researchgate.net/publication/255599452\\_On\\_the\\_Design\\_and\\_Implementation\\_of\\_User-friendly\\_Interface\\_for\\_Scientific\\_and\\_Engineering\\_Applications](https://www.researchgate.net/publication/255599452_On_the_Design_and_Implementation_of_User-friendly_Interface_for_Scientific_and_Engineering_Applications). [Accessed 14 October 2022].
- [12] A. Shi, F. Huo and D. Han, "Role of Interface Design: A Comparison of Different Online Learning System Designs," *Frontiers in Psychology*, pp. 1-9, 2021.
- [13] Bubble Team, "About Bubble," Bubble, [Online]. Available: [www.bubble.io/about](http://www.bubble.io/about). [Accessed 3 January 2023].
- [14] R. Guntupalli, "User Interface Design - Methods and Qualities of a Good User Interface Design [dissertation/master's thesis]," University West, Trollhattan, 2008.





# Appendix

The application being referenced can be found using this link: <https://nasm-x8664-assembly-app.bubbleapps.io/version-test/>