

EASY DB

By Austin Dantz, Spencer Pelton, William Watkins

Submitted to
The Faculty of the School of Information Technology
In Partial Fulfilment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

© Copyright 2020 Spencer Pelton, Austin Dantz, William Watkins

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Spencer Pelton

Spencer Pelton – Project Manager

Austin Dantz

Austin Dantz – Software Developer

William Watkins

William Watkins – Security and Testing

Abdou Fall

Abdou Fall – Technical Advisor

Date

04/17/20

University of Cincinnati

College of

Education, Criminal Justice, and Human Services

April 2020



~~~~~

Prepared by:

Spencer Pelton, Project Manager and Front-End Developer  
Austin Dantz, Software Developer  
and William Watkins, Security and Testing

Students of  
University of Cincinnati  
College of Education, Criminal Justice, and Human Services  
School of Information Technology  
[cech.uc.edu/it](http://cech.uc.edu/it)

April 2020

# Table of Contents

|                                                      |    |
|------------------------------------------------------|----|
| List of Illustrations .....                          | v  |
| Acronyms and Abbreviations .....                     | v  |
| Abstract .....                                       | 1  |
| <b>1.1 Introduction</b> .....                        | 2  |
| <b>1.2 Problem</b> .....                             | 2  |
| <b>1.3 Solution</b> .....                            | 3  |
| <b>1.4 Project Goals</b> .....                       | 3  |
| <b>1.5 Overview</b> .....                            | 3  |
| 2. Discussion .....                                  | 4  |
| <b>2.1 Project Concept</b> .....                     | 4  |
| <b>2.2 Design Objectives</b> .....                   | 4  |
| <b>2.3 Methodology</b> .....                         | 5  |
| <b>2.5 Use Case Diagram</b> .....                    | 7  |
| <b>2.6 Technical Discussion</b> .....                | 9  |
| 2.6.1 Application .....                              | 9  |
| 2.6.2 Database .....                                 | 9  |
| 2.6.3 Security .....                                 | 9  |
| 2.6.4 Web Application Screenshots and Examples ..... | 10 |
| 2.6.4.1 Login Page .....                             | 10 |
| 2.6.4.2 Landing Page .....                           | 10 |
| 2.6.4.3 Tutorials Page .....                         | 11 |
| 2.6.4.4 Tutorial Output Page .....                   | 12 |
| 2.6.4.5 Database Templates Page .....                | 14 |
| <b>2.7 Testing</b> .....                             | 15 |
| 2.7.1 Overview .....                                 | 15 |
| 2.7.2 Objective .....                                | 15 |
| 2.7.3 Entry and Exit Criteria .....                  | 15 |
| 2.7.4 Logging Test and Reporting .....               | 16 |
| 2.7.5 System Testing .....                           | 16 |
| 2.7.5 System Testing .....                           | 16 |
| 2.7.6 Pass/Fail Conditions .....                     | 17 |
| 2.7.7 Encountered Problems .....                     | 17 |

|                                    |    |
|------------------------------------|----|
| 2.7.8 Testing Plan Format .....    | 18 |
| 2.7 Budget .....                   | 19 |
| 2.7 Project Schedule/Gantt.....    | 21 |
| 2.8 Problems and the Future.....   | 22 |
| 2.8.1 Problem.....                 | 22 |
| 2.8.1 Future Recommendations ..... | 22 |
| 3.0 Conclusion .....               | 23 |
| 3.1 Fall Semester 2019 .....       | 23 |
| 3.2 Spring Semester 2020 .....     | 24 |
| APPENDIX A. REFERENCES.....        | 25 |
| APPENDIX B. POSTER .....           | 26 |

## List of Illustrations

### Tables

| <b><u>Section</u></b>                        | <b><u>Page</u></b> |
|----------------------------------------------|--------------------|
| <b>TABLE 1.</b> User Profile.....            | <b>6</b>           |
| <b>TABLE 2.</b> Testing Plan Format .....    | <b>18</b>          |
| <b>TABLE 3.</b> Project Budget Original..... | <b>19</b>          |
| <b>TABLE 4.</b> Project Budget Final.....    | <b>20</b>          |

### Figures

| <b><u>Section</u></b>                                   | <b><u>Page</u></b> |
|---------------------------------------------------------|--------------------|
| <b>FIGURE 1.</b> Use Case Diagram .....                 | <b>7</b>           |
| <b>FIGURE 2.</b> User Interface Diagram .....           | <b>8</b>           |
| <b>FIGURE 3.</b> Login Page.....                        | <b>10</b>          |
| <b>FIGURE 4.</b> Landing Page.....                      | <b>11</b>          |
| <b>FIGURE 5.</b> Tutorials Page.....                    | <b>12</b>          |
| <b>FIGURE 6.</b> Tutorials Output Page.....             | <b>13</b>          |
| <b>FIGURE 7.</b> Database Templates Page.....           | <b>14</b>          |
| <b>FIGURE 8.</b> Project Schedule and Gantt Chart ..... | <b>21</b>          |
| <b>FIGURE 9.</b> Poster .....                           | <b>26</b>          |

## Acronyms and Abbreviations

# Abstract

According to Marketing Metrics<sup>[1]</sup> businesses have a 60 to 70 percent chance to sell to existing customers versus just 5 to 20 percent for other prospects. A database can provide essential insights into a customer's choices that can be used to grow a business. Unfortunately, many small businesses do not have the knowledge or budget to create and manage a database. EasyDB aims to alleviate the stress on small businesses by providing a tool to learn the basics of database design as well as getting a template database created in real time that a small business can set up quickly with some acquired knowledge. This is done using EasyDB's guided template builder that walks users through the creation of essential database designs with the intention of creating an end database that the user can download for their own use. When the guided learning databases are not enough there are also templates, that users can edit and download to meet their specific needs. In the case that we tested with a small business owner, the owner was able to step through all of the guides and gain enough knowledge to implement a database that he could use in the future.

# 1. Problem Statement

## 1.1 Introduction

Data storage is a complex task that technical individuals have been implementing solutions to for decades. Due to solutions being created by technical individuals the most common applications require a technical background to implement the storage solutions. The amount of home businesses is growing each year, and most do not have the technical background to enact long term storage solutions or the budget to pay for a professionally managed storage solution. As a result, many small businesses are storing and analyzing data manually. While this solution addresses the problem of data storage, it is an inefficient time commitment that is unscalable for a growing business. The purpose behind EasyDB is to create a web-based application that addresses these problems by reducing cost and creating an effective saleable data storage platform.

## 1.2 Problem

According to the Minority Business Development Agency, the average cost for starting a business from scratch is \$30,000<sup>1</sup>. Starting a new business is not an easy task, and for non-technical business owners storing large quantities of data is daunting. Small Business Trends reports most non-technical business owners use spreadsheets instead of using databases or hiring technical employees to manage data<sup>2</sup>. Most owners do not have the time to learn how to code and manage databases which leads to the use of poor methods for tracking and storing data. Without technical assistance, owners have issues with manual data entry, managing inventory, manipulating data, and running reports for statistical insight. Alternative database and spreadsheet builders online are complex and do not ask simple questions or guide the user in the

creation process. Other solutions would require hours of user research and many different tutorial videos to learn.

## **1.3 Solution**

EasyDB aims to be an inclusive web application that provides users the ability to learn, create, and edit database essentials without needing prior database or coding knowledge. The application features a step by step guided tutorial that automatically builds a database for the user as well as providing template databases for the user to edit to their specific needs. This benefits small business owners by reducing the overall cost of database management by enabling them to create and manage their own database rather than outsourcing for their database needs. This also cuts the time needed to invest in learning as well as managing data storage by only teaching what is necessary for basic databases rather than overwhelming the users with database knowledge that is not a priority for small business.

## **1.4 Project Goals**

The purpose of EasyDB is to have a streamlined guide of essential database knowledge on an intuitive user interface design. This will aid in creating an acceptable learning environment in which users can quickly and easily navigate to what learning objectives they need for their specific database applications. By having a way to learn and implement a database quickly, we can meet our goals of cutting costs and cutting time investment for small businesses.

## **1.5 Overview**

In the remainder of this report we will cover the completion of the project. Including the following: Project concept, goals, testing methodology, budget, timeline, and concluding thoughts.

## 2. Discussion

### 2.1 Project Concept

The concept of EasyDB came from conversation with the IT staff at The University of Cincinnati College of Law. In passing a member of the IT staff in the College of Law mentioned they did not have much experience with database design and needed some database structures created for housing simple seating layouts for students for mock trials that are a common request by professors there. They had been requesting user information from the classes and manually creating “mini” databases that encompass a seating chart for rooms and classes each year. Although, there is a fix for this problem, the question we came out with was: How would a small business with no experience handle the creation of a database? We concluded that small businesses would be forced to either learn, outsource, or commit to manually storing data. Outsourcing can be out of budget for a small business and they may not have the time to commit to learning or storing data. This is when we came up with the concept of creating a web application that assisted in enabling small businesses to create and host a database in a time effective manner that only taught the essentials through guides and templates to cut time and cost on small business owners.

### 2.2 Design Objectives

The goal of EasyDB is to create a website for small businesses or individuals to quickly create and edit a database to their needs through template editor or guided tutorial that ends in a downloadable output file that can be uploaded to a database program locally. To do this, EasyDB must have an intuitive user interface to ensure a good learning experience as well as strong guided tutorials for users to learn from. Additionally, ensure there are usable templates for users to base their database structures on.

In development a few of our original goals had to be cut to ensure we maintained the focus of the project. At first, it was planned to have databases stored on site and to run prebuilt queries or saved queries on the databases. We abandoned these two goals because we came to an understanding that should the website gain traction, soon the cost of hosting the storage for free for many small businesses would require a payment plan to meet storage costs. This does not align with our goal of reducing cost on businesses, so this concept was abandoned in favor of local storage for individual cost and more freedom.

## 2.3 Methodology

The design methodology set for the project to help accomplish our goals were:

1. **Web Based** – The application was conceptualized as a website so that it would be easily accessible to non-technical users. This is used to ensure that users can begin the process of learning or designing database concepts quickly.
2. **Data Storage** - We knew from the beginning we would be dealing with databases and facilitating the ability of users to keep their costs low. To do this we decided to use XAMPP with MySQL because it is a free web hosting and database storage application to keep our costs low, which transfers to no cost to the end user. Also, we have tutorials on how to host the database once you export it so cost can be subjective to the storage needs of the business.
3. **User Interface** - We wanted a non-technical application that any user could use. This influenced our design procedure by having us test prototype builders on non-technical users to get feedback to confirm that the visual database builder is user-friendly.
4. **Must be able to export to other programs** - This influenced our decision on how to construct databases so that they export to other applications such as Excel or SQL.

## 2.4 User Profile

**Table 1: User Profile** presents the user profile used through the development of the product. It provides the team with some details of the audience or market we are creating the web application for.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>2.4.1 APPLICATION:</b><br>EasyDB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>2.4.2 POTENTIAL USERS:</b> <ul style="list-style-type: none"><li>- Small Business Owners</li><li>- Students</li><li>- Home accounting</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>2.4.3 SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:</b> <p>Users of EasyDB should have experience with web browsers, data interpretation, the ability to answer questions on personal data sets, and knowledge of data storage software such as excel, access, or SQLi.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>2.4.4 EXPERIENCE WITH SIMILAR APPLICATIONS:</b> <p>Some users may have experience with some of the following online and local solutions below.</p> <ul style="list-style-type: none"><li>- <a href="https://sqliteonline.com/">https://sqliteonline.com/</a></li><li>- <a href="http://sqlfiddle.com/">http://sqlfiddle.com/</a></li><li>- <a href="https://sqlzoo.net/">https://sqlzoo.net/</a></li><li>- Excel</li><li>- SQL</li><li>- SQLi</li><li>- Access</li></ul>                                                                                                                                                                                                                                                         |
| <b>2.4.5 TASK EXPERIENCE:</b> <p>Upon connecting to the web login, the user will be asked to fill a small survey to indicate what solution they may need. EasyDB will then suggest template databases as starting points for the user as well as a custom option to allow users to customize their own database from the ground up. During the creation phase, EasyDB will explain each database element and why it is used on the template to help indicate to the user where their data should go as well as giving suggestions to how tables should be linked. Upon completion the user will be prompted with optional export tracks such as excel, sql, or access and given instructions how to implement that export file.</p> |

#### 2.4.6 FREQUENCY OF USE:

This application is intended to be used once to create and export a database, but that template will be stored and can be used in help creating new queries that can be applied to the final database.

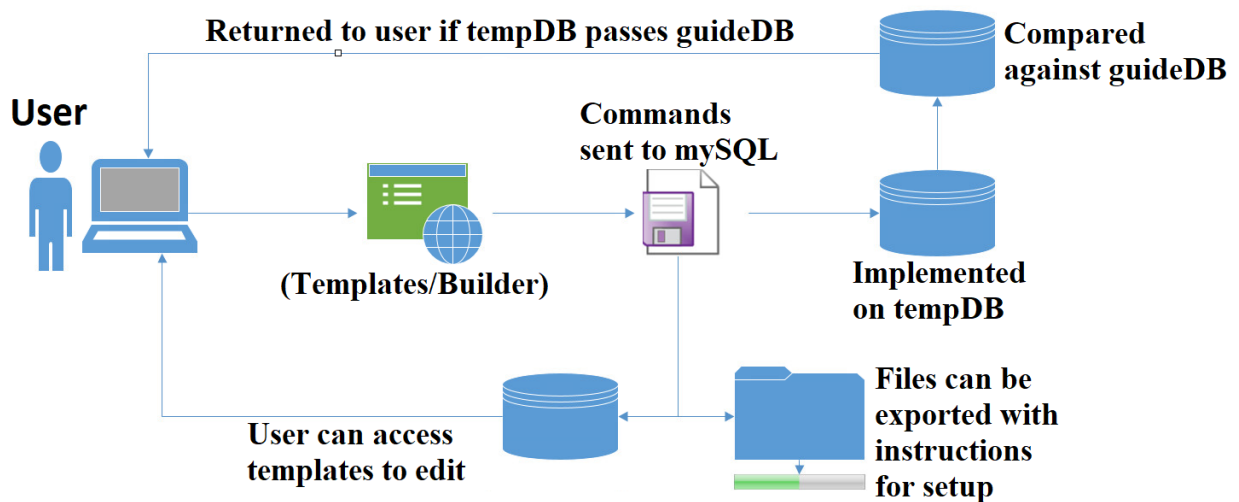
#### 2.4.7 KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Guided User Interface
- Dynamic visual representation of database template
- Simplified tile-based database storage and queries (Show/Run on click of tile)
- Guided export button (Describes how to implement exported database files)

**Table 1: User Profile**

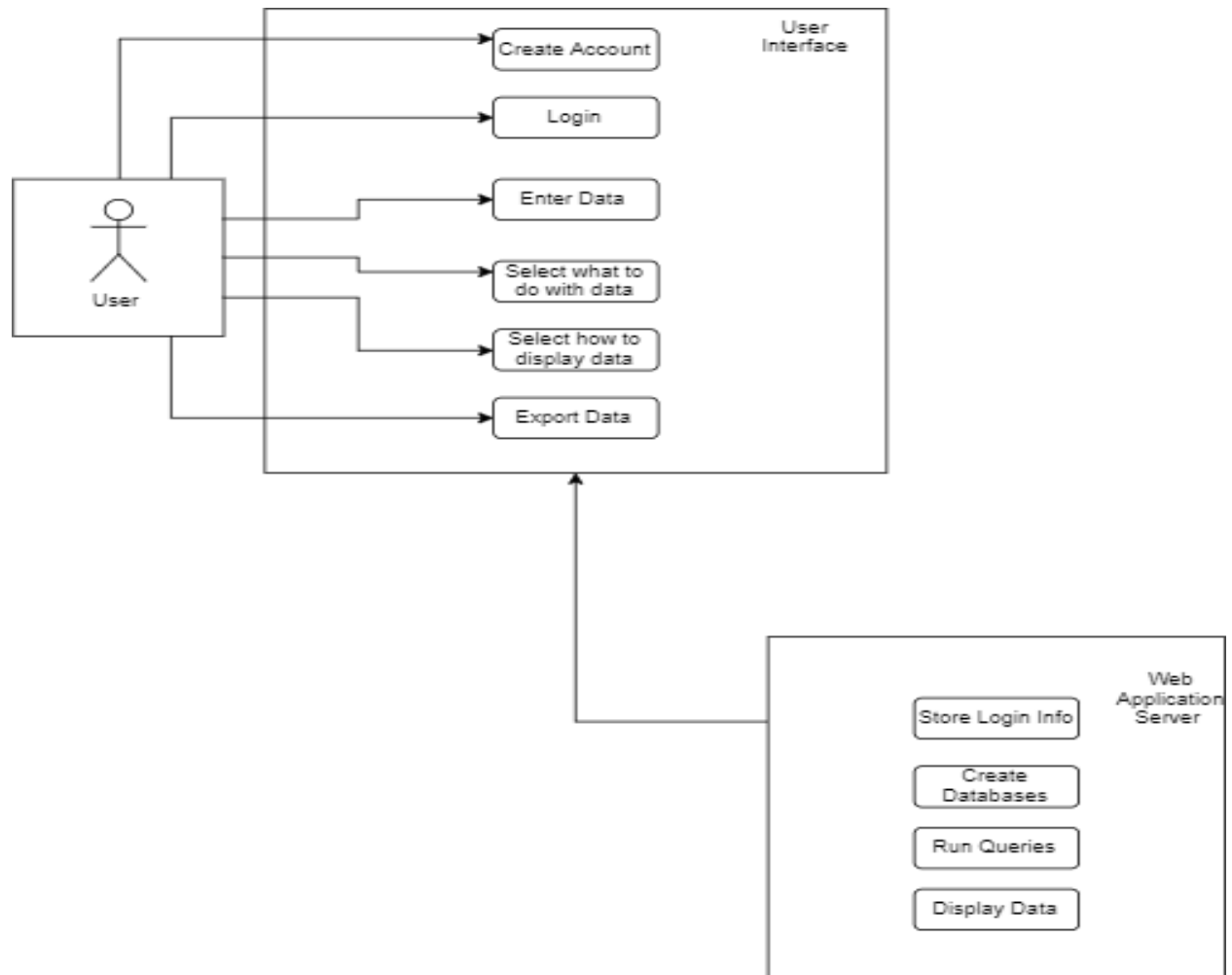
## 2.5 Use Case Diagram

**Figure 1: Use Case Diagram** presents the use case diagram. This represents the base level of requirements we have for our product. After these requirements have been met, more features will be added to better the customer experience.



**Figure 1: Use Case Diagram**

*Figure 2: User Interface Diagram* presents the user interface diagram. This represents the base level of requirements we have in producing our user interface. The basic visuals a user should interact with are shown below.



**Figure 2: User Interface Diagram**

## **2.6 Technical Discussion**

### **2.6.1 Application**

Our team utilized XAMPP as the hosting service for our web application as well as the built in MySQL database included in XAMPP. The database information that is easily transferred from the built-in database to the front-end website. Much of the code for the front-end user interface was written in PHP in Microsoft Visual Studio 2017. Github was utilized in controlling versions as well code sharing to test database templates. Although many versions are stored on Github, discord was also used occasionally for small updates. Unfortunately, XAMPP needs to be configured correctly with databases before the code can work properly to pull databases in some places of the code.

### **2.6.2 Database**

The application implements a variety of databases. First, we have the login database for users. Next, we have a tempDB and guideDB that are used in the creation of the tutorial database by creating a temporary database with an affix for the user that is then tested against the guideDB and deleted after comparison. Lastly, we have a few template databases that are stored both as plain text for immediate download as well as pre-built databases that can be edited for download.

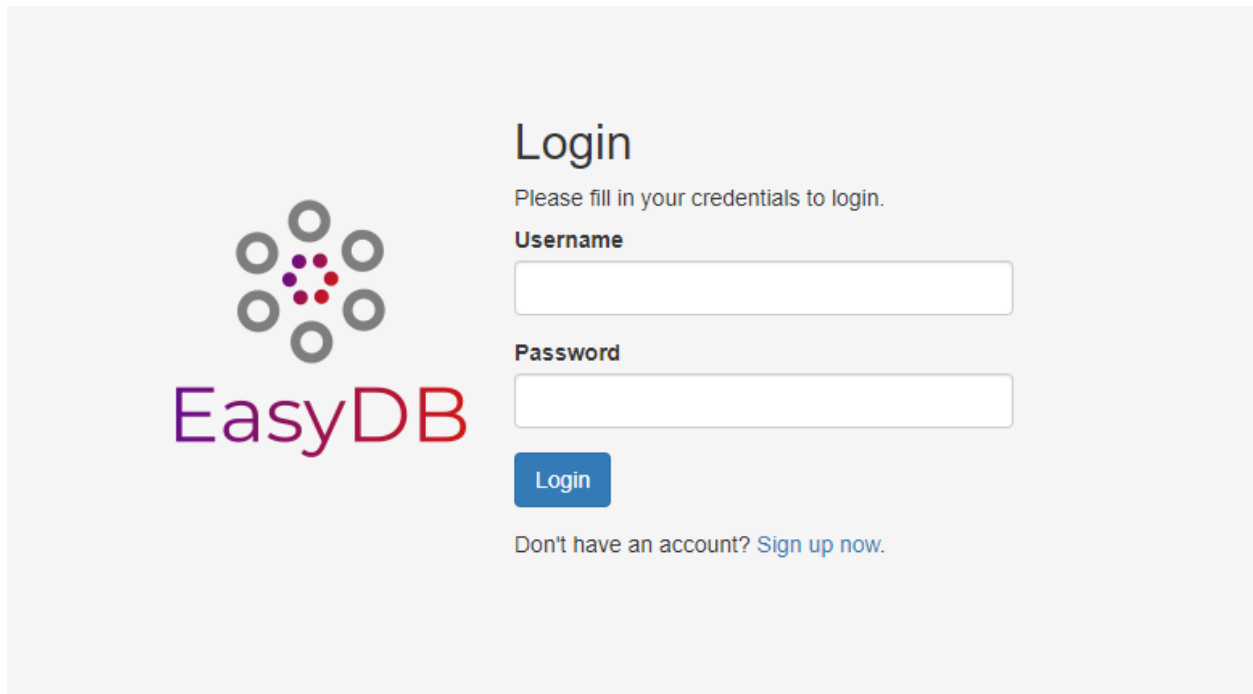
### **2.6.3 Security**

In the original design of EasyDB security had a larger role than it does now. Current applications do not require much thought towards security because data is stored locally for users. In the original design of EasyDB, where the databases would have been hosted on site, there would have to be consideration to user rights for small businesses that may have more than one user planning to add, remove, or view data. In either case, the login database must be encrypted to secure user's login data.

## 2.6.4 Web Application Screenshots and Examples

### 2.6.4.1 Login Page

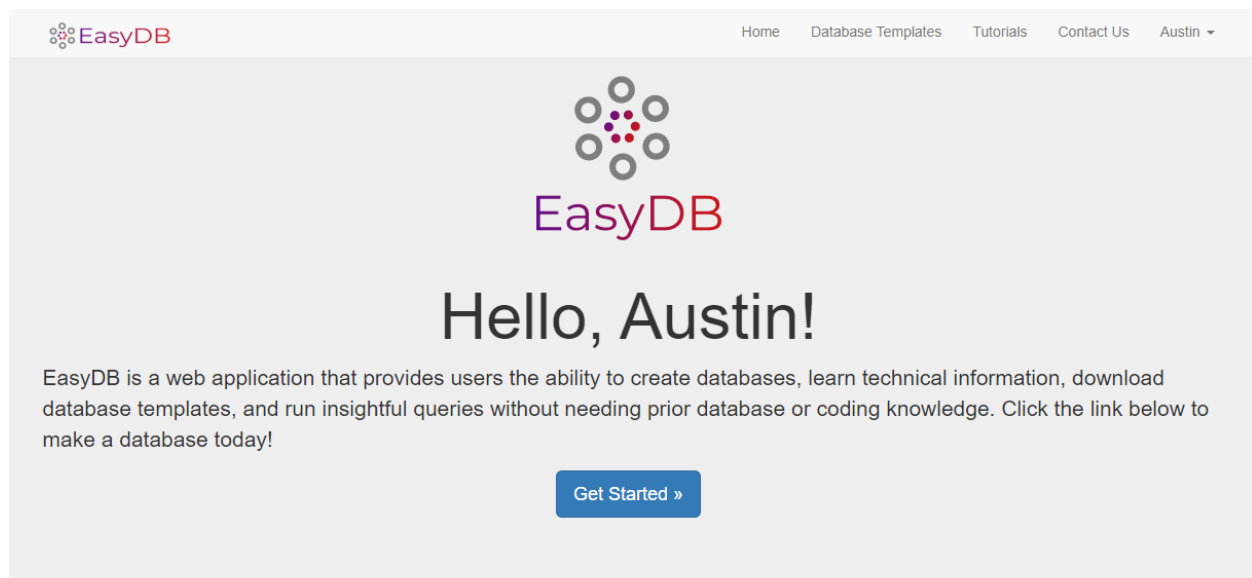
The Login Page is the first application page a user sees when opening the EasyDB website. The page contains a sign-in screen (*Figure 3: Login Page*) allowing the user to login with credentials. This will ensure the application loads the user's specific login data.



**Figure 3: Login Page**

### 2.6.4.2 Landing Page

The Landing Page is what the user sees after logging in to their account. It contains important information about what the application offers and displays the tabs Home (Landing Page), Database Templates, Tutorials, Contact Us, and User Profile (*Figure 4: Landing Page*). Selecting the Get Started button will take the user to the Tutorials page to create an example database. The page serves as a starting point for the application.



**Figure 4: Landing Page**

### 2.6.4.3 Tutorials Page

The Tutorials Page is shown to users after selecting the Get Started button on the Landing Page or selecting the tutorials tab along the top of the screen. The page allows users to select levels of learning and complete guided tutorials about learning databases (*Figure 5: Tutorials Page*). Once the tutorial is selected it will display a step by step guide for the level or lesson. The user will fill in SQL code along with the tutorial until it is finished.


[Home](#)
[Database Templates](#)
[Tutorials](#)
[Contact Us](#)
[Austin ▾](#)

Level 1: Creating a Basic Database

Level 2: Structure and Constraints

Level 3: Queries and Searches

© EasyDB 2019-2020


[Home](#)
[Database Templates](#)
[Tutorials](#)
[Contact Us](#)
[Austin ▾](#)

## Level 1 Tutorial: Basic Inventory Database

The first thing to do when working with databases is deciding on what the database is storing and then creating an appropriate name for the database. In this lesson we will be creating a database that will hold several items, the stock and description of these items, where the items are located, and where the items can be located.

This would be considered a basic Inventory database because it will be tracking the inventory of the stored products.

### Step 1.1: Naming

Since we know what we will be storing let's get started creating the database. First pick a name for the database that will appropriately describe what the database is storing.

A few examples: myInventoryDB, companyinventorydb, stockdb, etc.

Save this name as we will use it in the next step!

### Step 1.2: Creating the database

To create the database, you would type the following code below as a query and run it. **ENTER YOUR DATABASE NAME IN THE FIELD BELOW:**

CREATE DATABASE

**Note: All code typed will be run at the end of Level 1.**

### Step 2.1: The first table

Let's get started on creating a table. The database is an inventory database so the first table we know we need is a table that will hold data about an item in inventory.

When creating the table, a name for the table is needed as well as at least one column. In this case a name for the table and a name for the column will be needed.

The table name is a description of the table. Choose a name that will describe your inventory. **Note: You will enter this name in Step 2.2 code.**

A few examples: inventory, items, catalog, etc.

The Column name is a description of what data is going to be held. In this example we will want to store the names of items. Choose a name to describe your items column.

**Note: You will enter this name in Step 2.2 code.**

A few examples: Items, Units, Itemname, etc.

**Figure 5: Tutorials Page**

### 2.6.4.4 Tutorial Output Page

The Tutorial Output Page displays to the user their tutorial code, database tables created, and provide a local download of the database tutorial code (*Figure 6: Tutorials Output Page*).

## Congrats Austin on completing Level 1 and inventoryDB!

### Outputs

#### Code

```
CREATE DATABASE inventoryDB;

CREATE TABLE Items (Units VARCHAR(30) NOT NULL);

ALTER TABLE Items
ADD (item_description VARCHAR(30) NOT NULL)
ADD (stock int NOT NULL);

CREATE TABLE Locations (room_number VARCHAR(30) NOT NULL);

ALTER TABLE Locations
ADD (Enter the Column Name 'item_description'... VARCHAR(30));

ALTER Items
ADD itemID int;
ALTER Items
ADD PRIMARY KEY itemID;

ALTER Locations
ADD roomID int;
ALTER Locations
ADD PRIMARY KEY roomID;

CREATE TABLE LinkedTable (
  FOREIGN KEY (itemID) REFERENCES Items(itemID)
  FOREIGN KEY (roomID) REFERENCES Locations(roomID)
);
```

### Tables

Table 1: Items

| Units | item_description | stock | itemID |
|-------|------------------|-------|--------|
|       |                  |       |        |

Table 2: Locations

| room_number | Enter the Column Name 'item_description'... | roomID |
|-------------|---------------------------------------------|--------|
|             |                                             |        |

Table 3: LinkedTable

| itemID | roomID |
|--------|--------|
|        |        |

### Export

Austin, you can download the code and export it to any database program by clicking the button below:

[Download](#)

**Figure 6: Tutorials Output Page**

## 2.6.4.5 Database Templates Page

The Database Templates Page displays after being selected from tabs across the top of the screen. The page provides downloads to pre-created database code for any small business to use (Figure 7: Database Templates Page). They are common databases examples used every day. There is also an option on the Database Templates Page to Create your own Database, which is a step by step guide like the Tutorials Page, however it provides more flexibility to customize a database.

EasyDB

Home Database Templates Tutorials Contact Us Austin



**Multi-Location Orders**  
Product stock location and orders for a business with multiple storage locations. Use case for this database would be for a company that needed to store orders of their products and search for the stock of their products in separate rooms at separate locations. This database would also be able to store information about users and products and they're various locations.

Download



**Create your own Custom Database**  
This will open up a step by step questionnaire similar to the database tutorials. The difference is that you will be able to customize and create your own database this time with what you have learned from the tutorials. The use case is for someone who wants a truly unique database.

Get Building



**Basic Inventory Database**  
Holds simple information about products and they're location. A use case for this database would be for a company that solely needed to hold information about the products they carry and where they are storing those products.

Download



**Advanced Inventory Database**  
Holds information about products and they're location in multiple possible locations and the stock of items in that location. A use case for this database would be for a company that plans on holding information about products that may be stored in multiple stock rooms.

Download



**School Schedule**  
Holds information about teachers, students, classes, and class periods to create a class schedule for students and teachers. A use case for the database below would be for a business that connects users in a scheduled form similar to a school schedule where there are two entities that need to connect at a time and location.

Download



**Employee Information**  
Holds the information about employees, shifts and locations. A use case for the database below would be for a business that wants to store a schedule for their workers that display what shift the employee has and where they will be working that shift. Information is also stored about the employee for contact and what department they are in.

Download

Figure 7: Database Templates Page

## **2.7 Testing**

### **2.7.1 Overview**

This section covers the testing methodology used to thoroughly evaluate EasyDB's functionality and ease of use in all core areas of our project. The focus of the testing was to ensure our users have an easy, issue-free experience.

### **2.7.2 Objective**

The objective is to ensure that our users will have a easy, issue-free experience while using our site. To make sure that happens, we have set up a testing plan to check all of our core values. For example, we pride ourselves in the tutorials we provide to help teach our users; to make sure the tutorials are easy to follow and informative, we test them. We have testing regimes for each of our core values.

### **2.7.3 Entry and Exit Criteria**

Entry Criteria:

- Test Plan for section approved
- Testing Tools Ready (timer, charts, etc.-)
- Approved time of testing

Exit Criteria:

- All tests ran
- Bugs and failed tests documented
- Passed tests documented

### **2.7.4 Logging Test and Reporting**

While testing, if a bug or failed test occurred, document in high detail what happened and provide error codes and any other information gathered. Also record what was working well--not in high detail, just make note that the area was working well. Then, upload the information gathered to our shared document.

### **2.7.5 System Testing**

EasyDB was tested as a complete web application from various devices including cell phones, tablets, and computers. There was also internal testing from the device the site is hosted on. This allows us to check for bugs and fails from all devices that would be accessing our site. Each test conducted was tried on all devices.

### **2.7.5 System Testing**

The following are the testing procedures for each of our tests:

- Document prepared with information of what will be tested, what is needed, and the expected outcome
- Have organized tables ready to document pass/fails and to document bugs/ issues that need changed
- Record the outcome of the tests and upload to shared folder for others to review

Below are the tests that were performed:

1. Tutorials Testing - This test focused on the quality of our tutorials. We want our tutorials to teach our users easily, so this was tested by having family follow along with the tutorials and seeing how well it worked, taking their feedback, and having them try it again until it was acceptable.

2. Database Functionality – This test focused on the functionality of our database creation and manipulation--creating databases with all kinds of different data and running different queries we provide on said data to make sure it worked error-free.
3. Export Function – Our site allows you to export your data into Excel, MySQL, and Oracle. We need to test that the function works on all devices and all formats we provide.
4. Website Design Fluidity Functionality – This test aimed at making sure all the functions, buttons, links, images, etc. showed up well and worked on all pages of our site in a timely manner. This includes testing the site on different devices such as iPhone, tablet, and laptop, as well as different browsers.

### **2.7.6 Pass/Fail Conditions**

It is expected that EasyDB would pass all tests conducted, and considered successful if 5 tests in a row were successful, but if one did not pass the test, it is to be recorded in detail regarding what happened in order to be addressed.

### **2.7.7 Encountered Problems**

1. Exporting the data from the databases has been one of our bigger problems. While testing, we found that exporting to Oracle and Excel have the most issues; we get a lot of different syntax errors that needed to be worked through.
2. Another issue we found through having non-technical people try our tutorials is a UI flow problem. On step 4 of the first tutorial, we were having a few of our users get “lost” and end up confused. We ended up changing wording around and making a few visual aids to assist with that part.
3. Another issue that we found through non-technical users testing was that they would skip through the first tutorial or some parts of it, and then end up confused by the time they

were into the second tutorial because they skipped parts of the first one. To combat this, we track, and lock tutorials based on what has been completed.

## 2.7.8 Testing Plan Format

**Table 2: Testing Plan Format** presents the format used in testing and collecting outcomes to provide the team with information of passing and failing tests.

| Test Case                       | Steps                                                                                                                                    | Expected Outcome                                                        | Final Outcome                                                                                                                                       | Pass/Fail |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Export numerical data to Excel  | Take data provided for test and enter the data into the site, click through until you can choose to export, try to export to .xlsx       | Click export to Excel, then it downloads an .xlsx file to user's device | Clicked export to Excel and the file started downloading to my PC                                                                                   | Pass      |
| Export numerical data to Oracle | Take data provided for test and enter the data into the site, click through until you can choose to export, try to export to Oracle file | Click export to Oracle, then it downloads the file to user's device     | Clicked export to Oracle but it did not download to my PC, instead it showed syntax error – EOL while scanning string literal (screenshot included) | Fail      |

**Table 2: Testing Plan Format**

## 2.7 Budget

The original budget was planned regarding the potential costs of real-world needs should the website be published in its completed form. In the budget, the cost of an AWS Server meeting the minimum requirements as well as a copy of MySQL for the software and hosting needs was calculated. In addition to software and hosting needs labor was estimated at a projected hour of needed work. In practice of our final project, an AWS server was not need, nor a copy of MySQL. The largest mistake in our original budget was estimated work hours. It was clear once development began the hours were going to pass through the estimated hours quickly, particularly in SQL scripting and Website Development. The final budget reflects these changes including the new estimated hours worked and removal of cost of AWS and MySQL costs.

*Table 3: Project Budget Original* illustrates the original projected real-world cost for the development of the EasyDB website and application.

| EasyDB Virtual Environment Projected Costs |                          |             |            |                 |
|--------------------------------------------|--------------------------|-------------|------------|-----------------|
| NO.                                        | Item                     | Unit, Hours | Unit Price | Line item Total |
| Cloud Environment                          |                          |             |            |                 |
| 1                                          | AWS Server               | 1           | 175.93     | \$ 175.93       |
|                                            | Subtotal                 |             |            | \$ 175.93       |
| Software                                   |                          |             |            |                 |
| 2                                          | Windows Server 2016      | 1           | 110        | \$ 110.00       |
| 3                                          | MySQL                    | 1           | 1719       | \$ 1,719.00     |
|                                            | Subtotal                 |             |            | \$ 1,829.00     |
| Labor                                      |                          |             |            |                 |
| 4                                          | AWS Server Configuration | 4           | 25         | \$ 100.00       |
| 5                                          | MySQL Configuration      | 4           | 25         | \$ 100.00       |
| 6                                          | Website Development      | 40          | 40         | \$ 1,600.00     |
| 7                                          | SQL scripting            | 24          | 40         | \$ 560.00       |
| 8                                          | Reporting                | 8           | 40         | \$ 320.00       |
|                                            | Subtotal                 |             |            | \$ 2,680.00     |
|                                            | Total Cost of VE         |             |            | \$ 4,684.93     |

**Table 3: Project Budget Original**

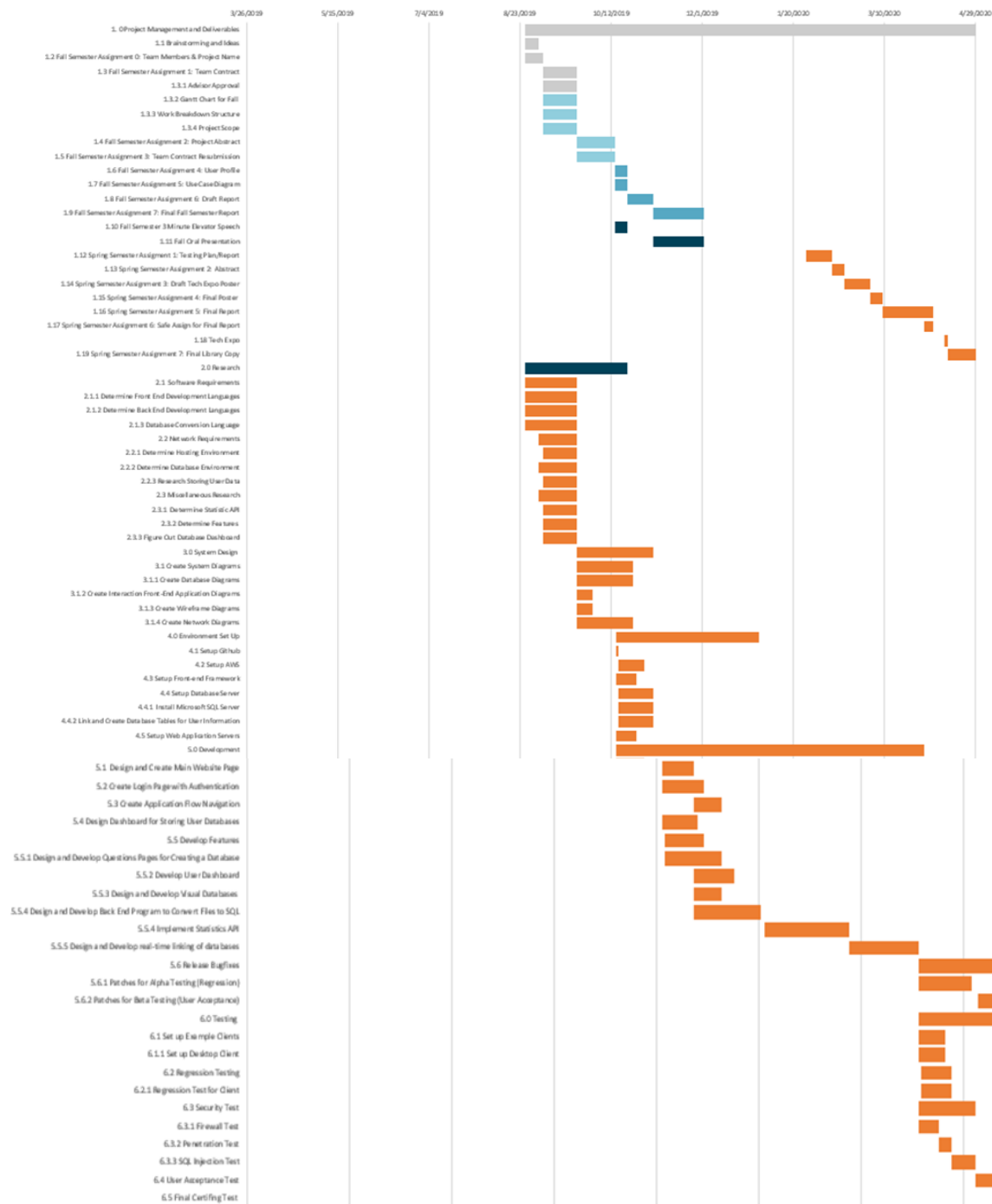
**Table 4: Project Budget Final** illustrates the final projected real-world cost for the development of the EasyDB website and application. This is based on our software used and experiences with creating the application.

| EasyDB Virtual Enviroment Projected Costs |                     |                |            |                 |
|-------------------------------------------|---------------------|----------------|------------|-----------------|
| NO.                                       | Item                | Unit,<br>Hours | Unit Price | Line item Total |
| Software                                  |                     |                |            |                 |
| 1                                         | XAMMP               | 1              | 0          | \$ -            |
| Labor                                     |                     |                |            |                 |
| 2                                         | MySQL Configuration | 4              | 25         | \$ 100.00       |
| 3                                         | Website Development | 90             | 40         | \$ 3,600.00     |
| 4                                         | SQL scripting       | 40             | 40         | \$ 1,600.00     |
| 5                                         | Reporting           | 12             | 40         | \$ 480.00       |
|                                           | Total Cost          |                |            | \$ 5,780.00     |

**Table 4: Project Budget Final**

## 2.7 Project Schedule/Gantt

**Figure 9: Project Schedule and Gantt Chart** is our project schedule with the milestones listed.



**Figure 9: Project Schedule and Gantt Chart**

## **2.8 Problems and the Future**

### **2.8.1 Problem**

During this project, the main recurring issue the group faced was through communication. Two group members are roommates which would communicate to each other frequently and that communication would not always translate to the third group mate. This was addressed a few times through creating a group message through app that had weekly recaps of progress, to-do lists, or reviews. In addition to this, due to the COVID-19 Crisis the group was not able to meet to effectively test the way that was originally envisioned of walking through guided learnings. Although the guided learnings were still tested, it was not through walk throughs with non-technical users outside of close members.

### **2.8.1 Future Recommendations**

For the future of EasyDB the best course of action for improvement would be to continue to create more detailed guided learnings for small business owners to learn from. As the project stands, the essentials of database management have been established and not much more as it was our goal to get a user to a working model as quickly as possible. With more in depth learning small business owners would be able to use their data more effectively. Another possible route for EasyDB would be to return to the concept of hosting data on site for those who do have the resources to spend and want to learn to control their data.

## 3.0 Conclusion

### 3.1 Fall Semester 2019

Iterating on ideas is important and it was a quick lesson we had to learn early on this fall semester. Our original idea for the project did not have enough substance and was lacking in technical depth. We had to overcome that idea and confer on a new project with the help of faculty, however, because of this experience I believe we learned well from it and our able to iterate on the project to meet goals. Communication is another lesson learned while working on the project. Managing and delivering a software application takes coordination and a lot of meeting to decide on deliverables. Realizing this we have been meeting up twice a week and organizing our project on a weekly basis with a lot of show and tell on objectives for the week. I believe it is the most vital lesson and skill we have enhanced / learned on the journey to creating this project.

Spencer learned PHP for front end development and realized how similar it was to JavaScript skills already developed. He expanded on his programming knowledge and branched out into a new language that suited project needs.

Will did not have much experience with version software control and was able to learn how to use GitHub thanks to other team members. He also gained experience with AWS when setting it up to be our hub for Microsoft Server 2016.

Austin learned how to create objects on a SQL server using scripts to find and edit databases.

While doing initial research on database scripting he enhanced his skills in Python and PowerShell.

In the Spring we plan to bring our database and web server together to test designs for user-friendliness through rigorous testing with non-technical individuals. Currently we will have most front-end development finished by the end of fall and initial database scripts. The spring will be mainly connecting the front end and back end with user testing.

### **3.2 Spring Semester 2020**

Spring Semester 2020 was primarily focused on cleaning up website design as well as filling in guided learning content and testing the website and tutorials for user friendliness. In this semester we faced uncertainty with the COVID-19 crisis that limited our options for non-technical user testing in the approach we had planned. In the end we tested our guided learnings and website UI on family members and house mates.

The lessons that resonates most from this semester is that timelines can change abruptly at a whim that require course correction to meet deadlines. Unlike any other semester expectations were changed quickly and you must be able to adapt and roll with the punches to succeed. In this semester with the COVID-19 crisis, classes were cancelled and presentations as well as expo were moved into online formats. To adapt to this, we spent time in video chats to simulate classes and practicing out presentation for the online format.

# APPENDIX A. REFERENCES

- [1] Farris, Paul, and Neil Bendle. Marketing Metrics. Marketing Science Institute, 2000
- [2] Beesley, Caron. "Minority Business Development Agency," 2011. [Online]. Available: <https://www.mbda.gov/news/blog/2011/11/how-estimate-cost-starting-business-scratch>.
- [3] Pilon, Annie. "Small Business Trends," 2013. [Online]. Available: <https://smallbiztrends.com/2013/06/what-is-a-database.html>.

# APPENDIX B. POSTER

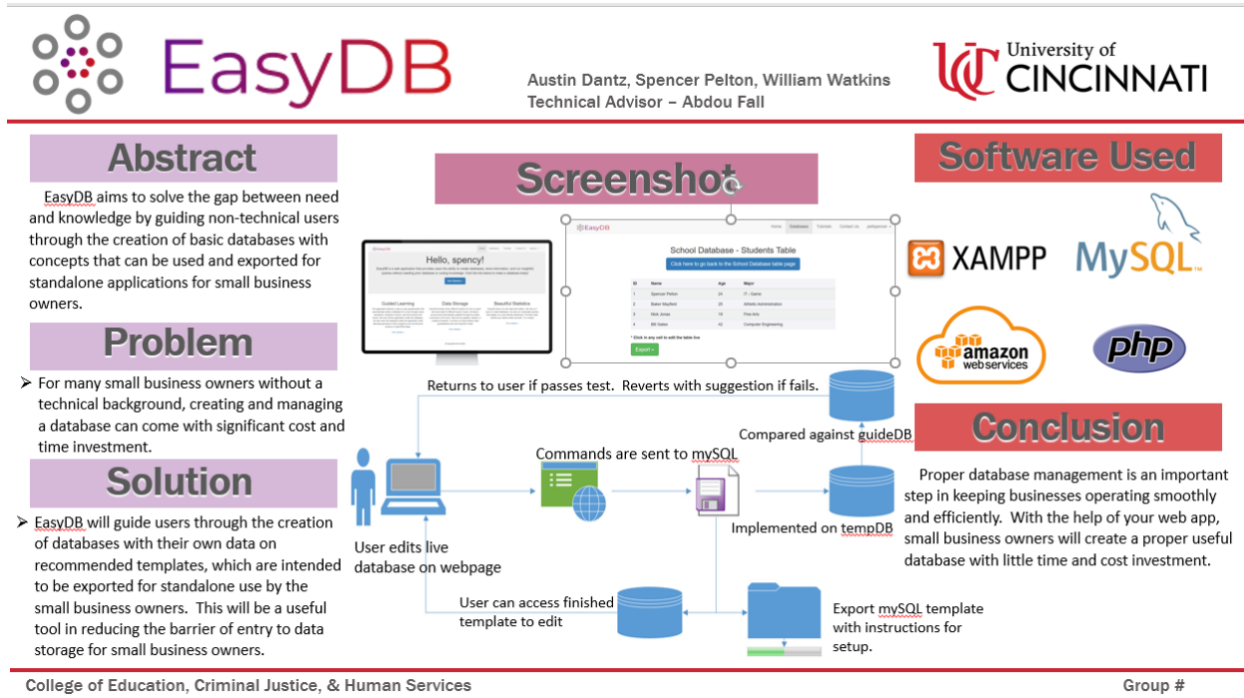


Figure 10: Poster