**Bubble.io Report**

**Introduction**

Bubble is a no-code full-stack application development web platform. This means that it provides a platform for users to create web application solutions from frontend to the database, all without writing any HTML, CSS, JavaScript, or SQL code. It does this by implementing a drag and drop system that feels more like graphic design than coding.

The database in Bubble.io can be set up either manually or by importing CSV files. However, the option to import a database via CSV is only available while using a paid plan.

**Parameters**

**Ease of Use**

Because Bubble is structured as a simple webpage that can be accessed from a browser, it does not need any other extra software to function. However, just like the MVHAC inventory system, it requires someone familiar with shapes and basic coding logic to create. For example, in MS Excel the various clickable buttons are created by formatting cells appropriately to the desired color and size. A similar thing is obtainable in bubble, except that there is a dedicated `button` function that opens a popup through which all the properties relating to the button can be set (such as color width and font). The same applies to setting up other elements in the homepages of the MVHAC inventory tool.

In MS Excel, the buttons perform actions based on macros while in Bubble the actions performed by the buttons and other clickable elements are based on ‘Workflow.’ The workflow defines what happens in response to user action (such as clicking a button) and can include storing the items in a form to a database or deleting specified items from a database (This is a major requirement).

To store whatever information the application developer wishes to store in on their website, Bubble provides the capability to create multiple databases. The structure of the database is the same with data stored in excel, except in nomenclature and the fact that Bubble database fields require that a field type (number, date, text etc.) be set. Interestingly, Bubble also allows a `list` field type that can accommodate a list of items as inputs. If MVHAC wants to store a list of all the items used to make an education kit in a database, the normalized database structure would require that a table be created for the individual items, another table be kept for the different kits, e.g. education, and that the two tables be joined using a foreign key. Bubble gives the option to avoid that kind of complexity by allowing a list of items to be added to a single field in any given record (rows in MS Excel lingo).

**Maintenance**

From briefly experimenting with Bubble, it is my opinion that after it is set up- by someone that knows something about basic programming logic and can effectively examine reference documentation found [here](https://manual.bubble.io/help-guides/getting-started)- It is not difficult to maintain a bubble application except as far as it is necessary to track monthly Workload units to determine when scaling is necessary. Workload Units will be discussed in the section addressing costs.

**Support**

Bubble has a support forum of users that can be found at <https://forum.bubble.io/>. Because it is a no-code platform, it is easier to find answers to concerns. I did a quick search on perplexity.ai regarding dedicated support by Bubble teams and got the following response that more concisely puts into words the results of the findings on this section:

“Enterprise Plan: The Enterprise plan offers unlimited priority support with a response time of 4 hours, making it suitable for complex and high-priority issues.

Starter and Growth Plans: These plans have a more standard support structure, with responses typically taking longer. The exact response times are not specified, but they are generally considered to be slower than the Enterprise plan.

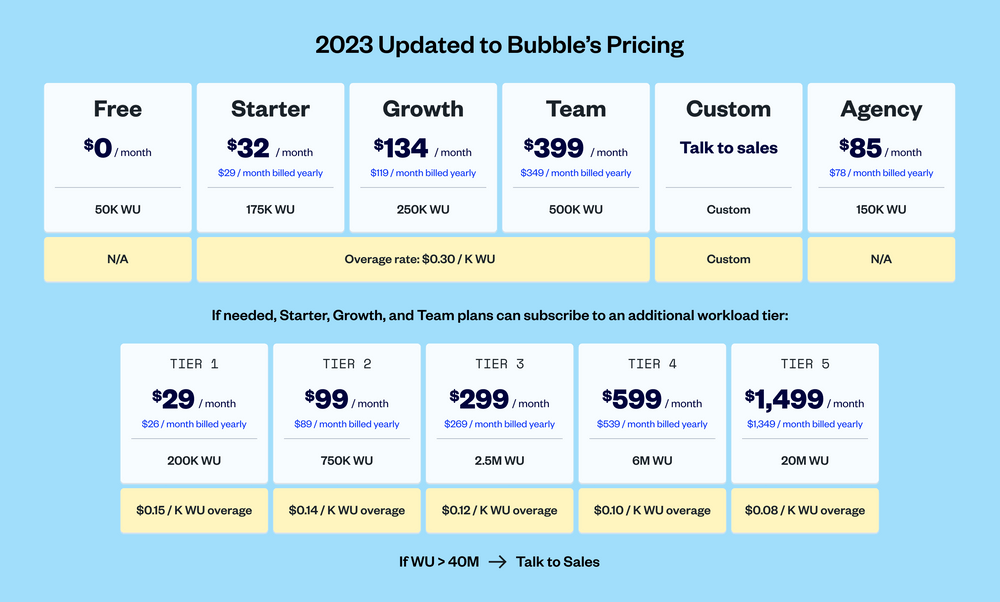
Free Plan: The free plan has limited support, with no specific response time guarantees.”

**Cost**

The diagram below explains bubble.io’s pricing plan. It is important to note that \workload Units (WU) refers to the amount of processing bubble.io’s back-end system has to do to keep the client’s application running. It is calculated by using the metrics outlined in [this post](https://manual.bubble.io/help-guides/workload/understanding-the-workload-calculation).

$3/100Gib/month for storage beyond what is offered in the plan MVHAC chooses to subscribe to

[Pricing plan](https://manual.bubble.io/account-and-marketplace/account-and-billing/pricing-plans)



As shown in the plan costs above, additional Workload units can be purchased at discounted rates by purchasing tiers.

I would recommend that MVHAC begins with the Growth Plan as it offers versioning and logging for 2 weeks. This can be useful in tracking what volunteers at the center do within any two-week period. This plan also offers more readily available support from the bubble team

**Reporting**

For reporting purposes, bubble.io lends itself very well to MVHAC. The platform offers various plugins (e.g. ApexCharts.js) that can be used to construct charts and graphs of varying designs. Some of the plugins require subscriptions, some require a one-time payment, and others are free (ApexCharts.js is free). Additionally, just as in the MVHAC inventory dashboard (MS Excel file), data from the database can also be displayed in dashboard format.

**Mobile Access**

Because bubble is a web application, it can also be accessed on mobile devices and the experience can be very similar to browsing a regular webpage. To achieve this, it is important that the site designer implements designs that are adequately responsive to changes in screen sizes (guides on design can be found in manual.bubble.io). I found that bubble web applications tend to take a longer time to load on mobile devices than they do on personal computers.

**Customization**

Customization is adequate enough to fit any present or future needs of MVHAC, baring a major surge in technological needs of the Humanitarian center. The dashboards and application pages can be designed and customized to fit needs.

**Backup**

Backup can be scheduled to an external backup facility. However, the frequency of backups and the scope of it contributes to the total cost per month.

**Replication**

The concept of replication is not there.

**Version Control**

The Growth Plan has logs and backups that allow the application to be restored to a prior state. As far as I could tell, there was no way of working on new features on a developer version of the MVHAC inventory app without affecting the production version.

**Security**

The security of data is dependent on bubble.io. MVHAC can configure a login or signup page to make certain that on approved users are granted access to the application. More granular controls an also be applied to make sure that MVHAC inventory users do not have access to data or authorizations that they do not need to perform their duties

**Integration**

Plugins can be used to gain access to the capabilities of various tools. The database can be exported to a third party.

**Scalability**

If the client’s needs exceed what the client paid for, bubble.io charges $0.30 per extra Workload units consumed. This helps ensure that the site stays online for its users even when a surge in usage pushes workload beyond initial expectations. the paid plans, the client can contact the bubble.io sales team to negotiate a custom plan to

**Resilience**

[perplexity.ai] Results indicate that Bubble has a strong community and support system to help users troubleshoot and resolve issues. Additionally, Bubble provides built-in backup and error handling features to mitigate the impact of crashes.

Overall, while Bubble apps may experience occasional performance or stability issues, they do not appear to crash regularly for most users. With proper planning, testing, and utilization of Bubble's features and community support, Bubble app builders can minimize the risk of crashes and ensure a reliable user experience.