



VIDHYADEEP INSTITUTE OF COMPUTER & INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the seminar , submitted for the Seminar entitled “**Low/No Code Development**” has been carried out by “**Darshan A Nariya**” (E19110403000110036) at BCA Department of Vidhyadeep Institute of Computer & Information Technology, Kim for partial fulfilment of BCA degree to be awarded by Veer Narmad South Gujarat University.

Place:- KIM

Date:-

Internal Guide

Head of Department

JURY

Index

CERTIFICATE	I
Index	II
1 Abstract.....	1
1.1 Background.....	1
1.2 Aim.....	1
1.3 Method.....	1
1.4 Result.....	1
1.5 Conclusion.....	1
2 What is Low/No Code Development	2
3 Why LCD	3
4 History of LCD.....	4
5 Types of LCD	5
5.1 Low code	6
5.2 Drag and Drop	8
5.3 Visual Programming.....	10
5.4 Pre-made Packages.....	12
5.5 WYSIWYG	13
5.6 AI Assist	15
6 Advantages	17
7 Disadvantages.....	18
8 Platforms.....	19
9 Uses.....	21
10 Conclusion	22
11 References.....	23

1 Abstract

1.1 Background

In recent years, Low-code development (LCD) is growing rapidly, and Gartner and Forrester have predicted that the use of LCD is very promising. Giant companies, such as Microsoft, Mendix, and Outsystems have also launched their LCD platforms.

1.2 Aim

In this work, I explored two popular online developer communities, Stack Overflow (SO) and Reddit, to provide insights on the characteristics and challenges of LCD from a developers and Industry perspective.

1.3 Method

I used two LCD related terms to search the relevant posts in SO and extracted many posts. Meanwhile, I explored three LCD related subreddits from Reddit and collected many posts from there too.

1.4 Result

Findings show that LCD may provide a graphical user interface for users to drag and drop with little or even no code, the equipment of out-of-the-box units (e.g., APIs and components) in LCD platforms makes them easy to learn and use as well as speeds up the development, LCD is particularly favoured in the domains that have the need for automated processes and workflows; and developers have conflicting views on the advantages and disadvantages of LCD.

1.5 Conclusion

Findings suggest that researchers should clearly define the terms when they refer to LCD, and developers should consider whether the characteristics of LCD are appropriate for their projects.

2 What is Low/No Code Development

No-code implies that no hand-coding is necessary, while Low-code relies on no-code elements such as visual drag-and-drop components combined with some hand-coding.

Technically These two can be two different topics in their own, but because No code development isn't that vast enough so here, we will talk about Low Code Development with little bit of No Code Development.

Firms prefer to choose low-code alternatives for fast, continuous, and test-and-learn delivery. The survey performed by Forrester also shows that LCD platforms can accelerate development by 5 to 10 times. Moreover, these platforms also offer enterprises a more economical way to fulfil the market and/or enterprises internal requirements.

Although LCD is booming in industry, there is no clear understanding of LCD as well as its practices. To this end, we plan to explore the characteristics and challenges of LCD from the perspective of developers.

Research found that most developers tend to use low-code (e.g., “The coding effort is low”) to describe LCD. In other words, they think that the coding effort is low in LCD. The term drag and drop comes second, followed closely by visual programming.

Some developers also use pre-designed templates, non-professional programmers friendly, what you see is what you get (WYSIWYG), and business process to demonstrate their understanding and perception of LCD. A few others consider that LCD utilizes a graphical user interface to develop programs, and one use case is built automation to “automate unattended operations with minimal human involvement”.

3 Why LCD

With the growth of the Internet and the wave of digitalization, there is a growing need for enterprises to make quick and resilient responses to changing market requirements.

According to the research company Gartner, the demand for information systems will increase five times faster than the ability to provide them by IT departments, because number of employees is not growing at a sufficient pace. Furthermore, recruiting software engineers has become increasingly difficult as demand is high and supply is low.

In order to solve the problems above and adapt to this rapidly evolving world, companies are looking for quicker and cheaper ways to meet their software needs. In response, low-code development (LCD) platforms have emerged with the promise that organizations can hire business professionals with no coding experience to build applications.

The main goal of LCD as it suggests is to make development require less coding or less coding experience.

The main goal of LCD as it suggests is to make development require less coding or less coding experience.

4 History of LCD

While the term “Low code development” is relatively new, the actual method it works is not that new.

Low-code development platforms trace their roots back to Fourth-Generation Programming Language (4GL) a concept that was developed from the 1970s through the 1990s, overlapping most of the development of 3GL.

- 3GL examples: C, C++, Java, Python, PHP, Perl, C#
- 4GL examples: Unix Shell, SQL

4GL made programming relatively easy and more human readable which made it more developer friendly and required less developer effort than 3GL languages.

So, any technology which made a tiresome task easy can be considered a step forward in Low Code Development.

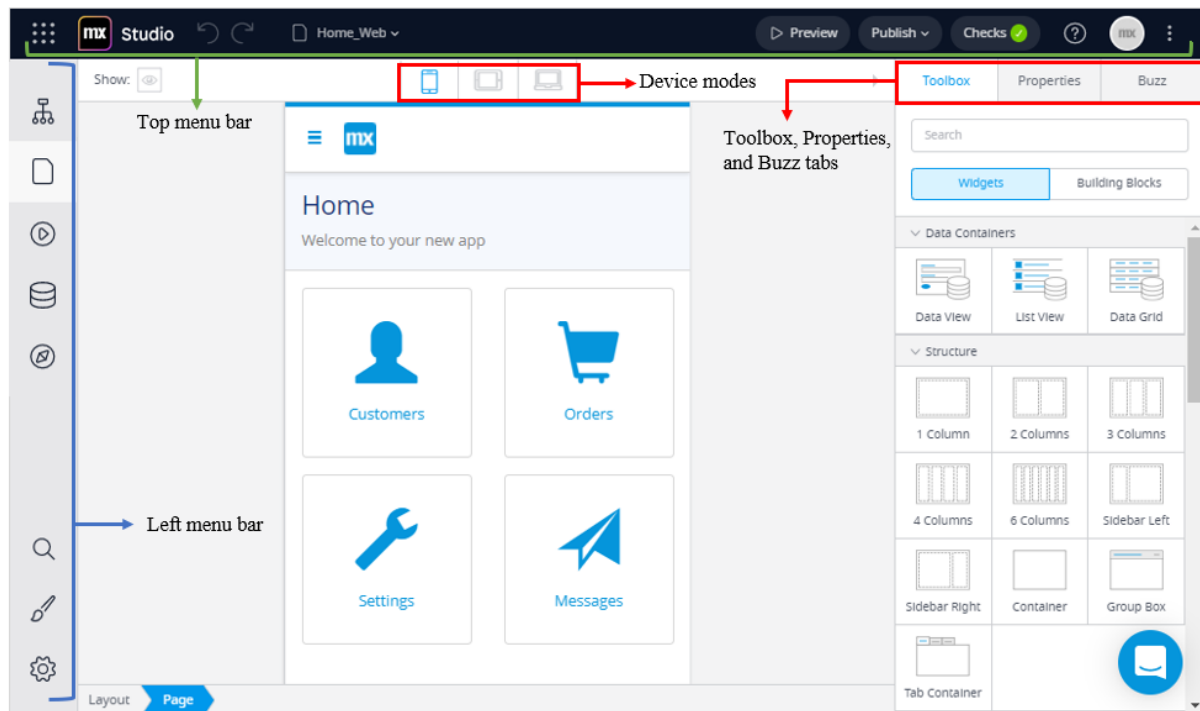
5 Types of LCD

For now, there is no standard definition for LCD in industry according to my research, and there are no predefined types of LCD so here I have divided them by the methods it uses to explain its type accordingly.

Term	Description
Low code	You need less programming skills and you are able to realize your processes without the need of coding.
Drag and drop	Can probably do everything through drag-and-drop.
Visual programming	low-code is a visual approach to software development.
Pre-made packages	Pre made packages or templates to ease tasks.
WYSIWYG	Meant for What You See Is What You Get app maker.
AI Assist	Artificial Intelligence Assisted Programming

5.1 Low code

Low-code is a visual approach to software development that optimizes the entire development process to accelerate delivery. With low-code, you can abstract and automate every step of the application lifecycle to streamline the deployment of a variety of solutions.



The term “Low code” is often used for solution that provide everything for developing an or S/W from start to the end like Creatio and Mendix. These two examples are commercial and do not provide open-source availability so developers who want control don’t prefer them while it is great choice for non-developers with little to no knowledge of programming.

Advantages

- Low-code solutions provide all facilities needed for developing an app.
- It is User Friendly.
- No coding knowledge needed.

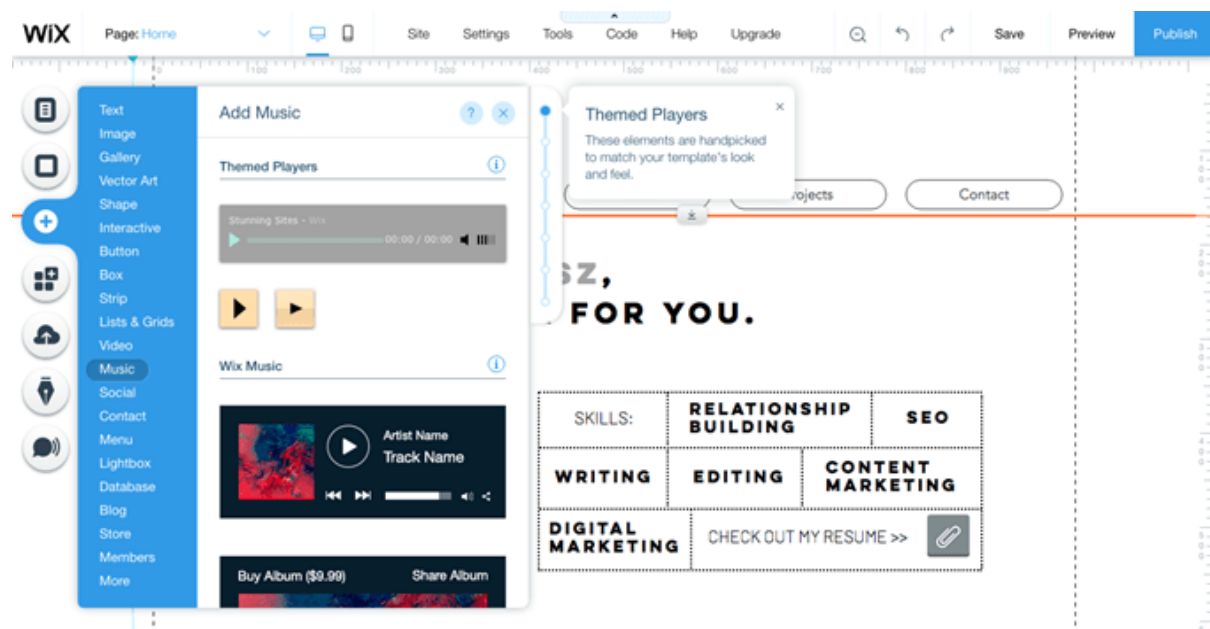
Disadvantages

- Not all solutions are open source.
- User still have to be technical to understand workflow of Low-code.
- It doesn't provide control over all things.

5.2 Drag and Drop

The term “Drag and drop” as it suggests is used for solutions that offer drag and drop feature to make applications. Some solutions also provide drag and drop plus code for more flexibility.

This solution makes it extremely user-friendly cause user can just drop components and arrange them according to their liking, without coding a single line. Many Website builders fall under this category like Wix.com, WordPress, Microweber and CMS hub etc.



While drag and drop is easy for beginners and non-experienced not all of them provide full-fledged solutions and there comes in drag and drop plus code where custom code can be implemented with drag and drop components. Like Visual Studio, WordPress etc.

Advantages

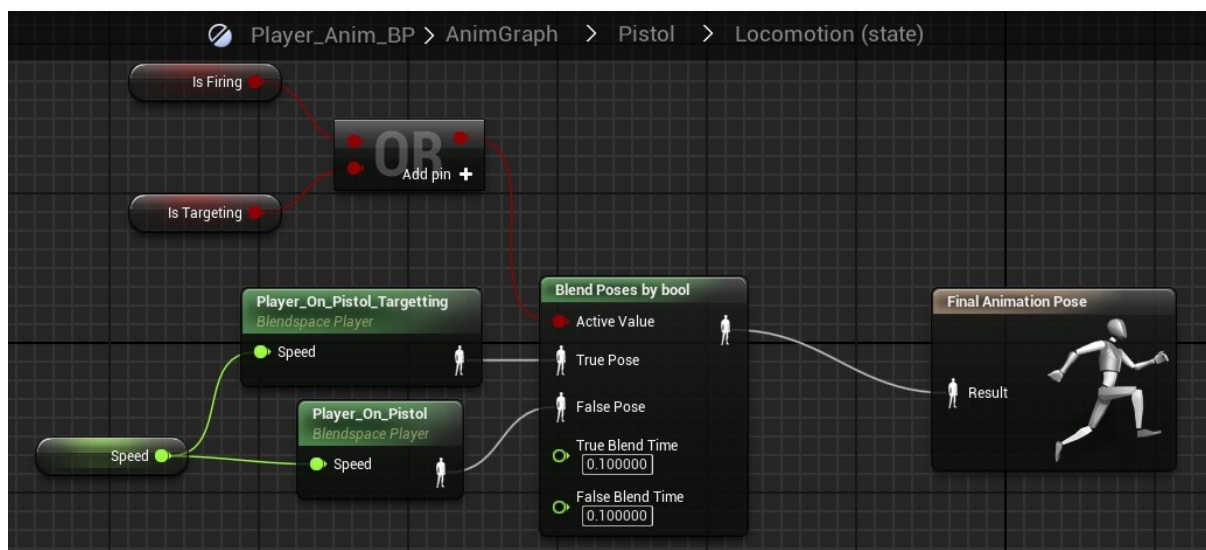
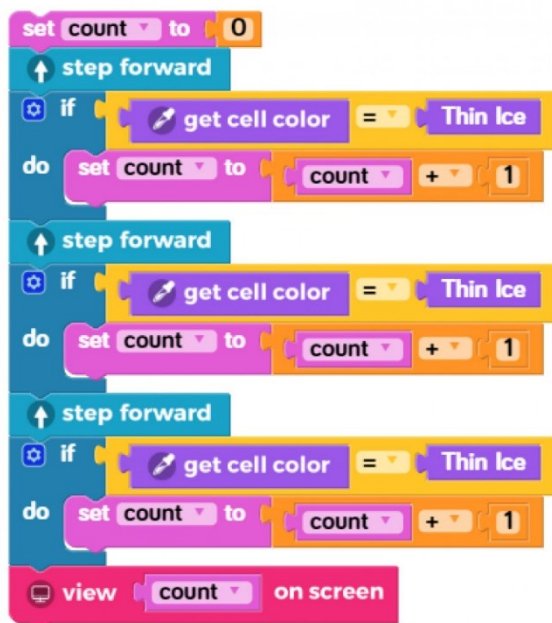
- Extremely user friendly.
- With drag and drop plus coding it makes building apps very easy.

Disadvantages

- All platforms don't provide custom coding functionality.
- Most of solutions are commercial.

5.3 Visual Programming

The term “Visual Programming” is used for visualized code where instead of blocks of code it is showed in parts of code is showed in form of building blocks or nodes and you can change or add values inside these blocks manually. this method makes it beginner friendly and different from commonly used languages. like Scratch, Blockly etc.



While this method is mostly used in teaching kids or newbies programming but it is widely used in 3d modelling and games studios like Unreal, blender, Unity. Nodes based visual programming is difficult in its own and needs understanding to use but many people in community have made fabulous things out of it.

Advantages

- Many Visual programming languages are easy to learn.
- Languages provide many built-in objects that can be used to make applications.

Disadvantages

- While they are easy to learn they are hard to master.
- After a point visual programming code becomes hard to manage.

5.4 Pre-made Packages

This term “Pre-made Packages” also self-describe itself; it means using already made packages provided by community or solution provider.

These packages can be pre-installed or can be installed according to our need, there are many open-source package managers in community like npm, composer, yarn, etc.



Most of the time one package only provides few functionalities so we have to install multiple of them to full-fill our needs, while most of the package managers provide easy management one still needs to have fair enough programming knowledge, but if one has the necessary knowledge packages makes so many things easy.

Advantages

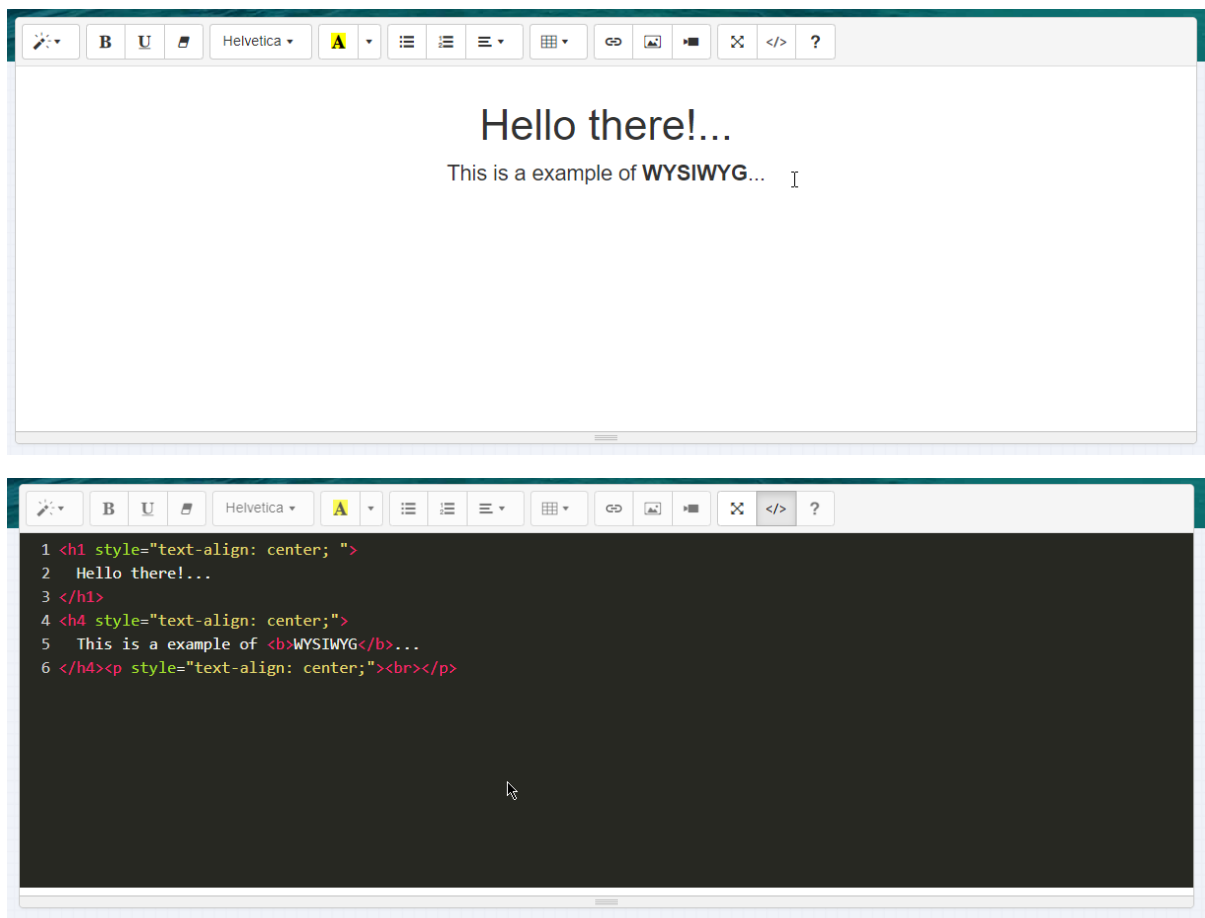
- We don't have to program every single feature.
- There are many free and Open source packages available.

Disadvantages

- Package managers can be learning barrier to beginners.
- Not all packages are trustable.

5.5 WYSIWYG

The term “WYSIWYG” stands for “What You See Is What You Get” another one which explains itself, WYSIWYG is type of system where user makes page like a word document and solution provides a web page based on what you entered, like Kompozer, Dreamweaver, etc. Text editors used inside web browsers like CKEditor, summernote are very known example of this too.



WYSIWYG is relatively old now because in most cases it only provides tag formatted code, so it doesn't complete whole job but it makes it easier, WYSIWYG can still be seen in web solutions where user have to commonly change content there, we can use WYSIWYG Editor to make things easy.

Advantages

- These are simple and easy to use.

- These generate accurate code most of time only design problem occurs in old systems.
- Implementing these in applications makes CMS easy.

Disadvantages

- Code generated by these can be bulky.
- Not all editors follow W3C standard.

5.6 AI Assist

“AI Assist” means help of Artificial Intelligence to do coding instead of manual coding, before AI assist, we have seen code editors suggest code snippets while we write some common code, but they only suggest basic syntax.



That's where AI assist comes in, AI assist doesn't just suggest syntax it suggest entire blocks of functions based on what you have already written in your code. There are many of these AI assist plugins out there like, GitHub co-pilot, Kite, JetBrains, etc. here I have shown example of GitHub co-pilot

```

837     $data = compact( 'title', 'menu' );
838     return view('AdminPanel.chng_profile.form', $data);
839 }
840
841 public function save_profile(Request $request)
{
    $request->validate([
        'name' => 'required',
        'email' => 'required|email',
        'image' => 'image|mimes:jpeg,png,jpg,gif,svg|max:2048'
    ]);

    $id = $request->session()->get('AdminUser')['id'];
    $user = User::findOrFail($id);

    $user->name = $request->name;
    $user->email = $request->email;

    if ($request->hasFile('image')) {
        if ($user->image) {
            Storage::delete('public/userdata/' . $user->image);
        }
        $image = $request->file('image');
        $name = time() . '.' . $image->getClientOriginalExtension();
        $destinationPath = public_path('/userdata');
        $image->move($destinationPath, $name);
        $user->image = $name;
    }

    $user->save();
    $request->session()->forget('AdminUser');
}

```

Advantages

- Helps suggesting entire blocks of code based on code you have written or name of your file and extension.
- More you use more it becomes accurate.

Disadvantages

- It doesn't suggest accurate code all the time.
- You need programming knowledge to know what AI is suggesting.
- These technologies are still in development.

6 Advantages

Build applications faster with minimal effort

It makes development more agile as LCD platforms are equipped with rich and ready-to-use implementation units.

Easier than programming

Many LCD platforms are easy to study and use compared to programming languages.

Make apps without need of developers

LCD reduces the time required for development and allows non-professional to implement their ideas without having to hire developers.

Newbie friendly

Many LCD have drag and drop features which means anyone who has decent knowledge of word can make apps on them.

7 Disadvantages

Vendor lock-in

Some LCD solutions are hard to modify, maintain, and debug. The potential reason could be that it is difficult to verify whether the implementation units provided by LCD platforms have defects due to no access of code, and different LCD platforms may bring different experiences to developers.

Learning curve

LCD is indeed easier than programming, but the use of LCD platforms also has a certain learning cost, especially some LCD platforms provide complex functions, which take time to learn.

Expensive

Some commercial LCD platforms also require a high price to provide a complete service, and some of the platforms charge for every user, which means that they can get very expensive as you scale your team

It limits experienced developers

Some LCD platforms are designed more like a prototyping tool and also target for non-developers which makes it very difficult for experienced developers to use.

Limited and Inflexible

LCD seems intuitive if the users only build applications with drag and drop operations. Once they find it limited and inflexible and have to add custom code at some point, LCD may bring complexities, compromises, and frustrations. Furthermore, if complex functions still need coding to achieve, it means that LCD is less powerful than programming.

8 Platforms

Below are some of popular platforms used for LCD , their company name and whether they are commercial or Open Source.

Platform Name	Company Name	Open-Source or Commercial
Bubble.io	Bubble	Commercial
Webflow	Webflow Inc.	Commercial
Adalo	Adalo	Commercial
Airtable	Airtable	Commercial
Appgyver	Appgyver	Open-Source
Glide	Glide	Open-Source
Wix (Editor X)	Wix.com Inc.	Commercial
Power Apps	Microsoft	Commercial
Zapier	Zapier	Commercial
DronaHQ	Deltecs Infotech Pvt. Ltd.	Commercial
Wordpress	Wordpress Foundation	Open-Source
Softr.io	Softr	Commercial
Backendless	Backendless	Open-Source
Appsheet	Google	Commercial
Outsystems	Outsystems	Commercial
Thunkable	Thunkable	Commercial
Draftbit	Draftbit	Open-Source
Xano	Xano Inc.	Commercial

Platform Name	Company Name	Open-Source or Commercial
Wappler.io	Wappler	Open-Source
Shopify	Shopify	Commercial
Integromat	Celonis GmbH	Open-Source

9 Uses

Common Application

Out of all applications of LCD frontend is the most used part where LCD is used then comes backend in which APIs are most to build by LCD.

Programming Languages used in LCD

To some degree, depends on the supporting technologies behind LCD, which are largely specific to the LCD platform chosen.

Java and JavaScript are both popular programming languages used for LCD, which is reasonable in that they are also popular languages in software development.

Types of Applications developed by LCD

Many posts state the types of applications which developers created or were creating using LCD. From the result, the demand for developing mobile applications with low code is the highest among the types of applications, which is reasonable since mobile applications normally have a short delivery time and their development can be sped up by employing LCD.

Different LCD platforms support the development of different types of applications. For instance, Bubble introduces a way to build web applications without code, Storyboard is used to create interfaces for iOS apps with zero code.

10 Conclusion

LCD is not a new concept in software development, but recently is booming and gets much attention in industry that can help both developers and users quickly deliver applications by writing a few code.

I used two LCD related terms to search the relevant posts in SO and extracted many posts. Meanwhile, I explored three LCD related subreddits from Reddit and collected many posts from there too.

- Although there is no clear definition of LCD, developers tend to use low-code and drag and drop to describe LCD according to their understanding, showing that LCD may provide a graphical user interface for users to drag and drop with little or even no code.
- The equipment of out-of-the-box units in LCD platforms makes them easy to learn and use as well as speeds up the development.
- Different LCD platforms support the development of different types (e.g., mobile and web applications) of applications and different parts (e.g., frontend, backend) of applications.
- LCD is particularly favoured in the domains that have the need for automated processes and workflows.
- While LCD platforms can speed up software development with minimal human involvement, they also suffer from no access of source code and vendor lock-in for commercial LCD platforms. Moreover, developers have conflicting views on the advantages and disadvantages of LCD, implying that certain features of LCD are beneficial to development if used appropriately, otherwise may become limitations or challenges in LCD.

11 References

Keywords

Low-Code Development, Stack Overflow, Reddit
“low-code”, “zero-code”, and “no-code”

Websites

<https://betterprogramming.pub/low-code-history-b756c095494f>

<https://www.outsystems.com/>

<https://stackoverflow.com/>

<https://www.reddit.com/r/lowcode/>

<https://www.reddit.com/r/nocode/>

<https://www.reddit.com/r/nocodelowcode/>