QWEB

Processing-Inspired Website Language

Our Team

Tamanna Hussain

Project Manager

Kamrul Hossain

Systems Architect

Ramisa Murshed

Language Guru

Xabi Peralta

Tester

Table of Contents

01 **Introduction**
Motivation, intended audience, use cases, etc.

02 **Features**
Data types, function / variable declarations, etc.

03 **Implementation**
Software technologies, architecture

04 **Future Work**
Additional features to implement

05 **Demonstration**
Demo of sample program

Introduction

QWEB is a **website language** inspired by interactive visualization languages such as Processing.js. Its purpose is to help both novel and experienced programmers **design and develop websites** to run in their browser

```
.n visKvartal(aar,

  ar ugeraekker = tabelUger.
 for (var ur=2; ur<ugeraekke:
 {
   var rk = ugeraekker[ur];
   rk.style.display = (rk.class)
 }
 }
</script>

<table style="font:8pt arial" width='
 <td>

   <h3>Olga Antoniuk</h3>

 </td><td align=right>Vis alle uger:

   <a href="Javascript:visKvartal
        2011 kv.3</a>

   <a href="Javascript:visKvart/
        2011 kv.4</a>

   <a href="Javascript:visKv/
        2012 kv.1</a>
```

Motivation

To create an alternative website language that is more intuitive, maximizes human readability, and incorporates familiar programming constructs that are used in traditional high-level programming languages

01

Language Features

Brief overview of data types, function and variable declarations, etc.

Language Features

Data Types

- String
- Integer
- Float
- Boolean
- Void

Variable Declaration

```
int a;

a = 50;

bool b;

str c;

c = "Hello World"
```

Language Features

Function Definition

```
function int check(bool b)
{
   int a;
   a = 3;
   if (b)
     print(a)
   output a;
}
```

Function Call

```
function int main(){
 print(check(true));
 print(check(false));
}
```
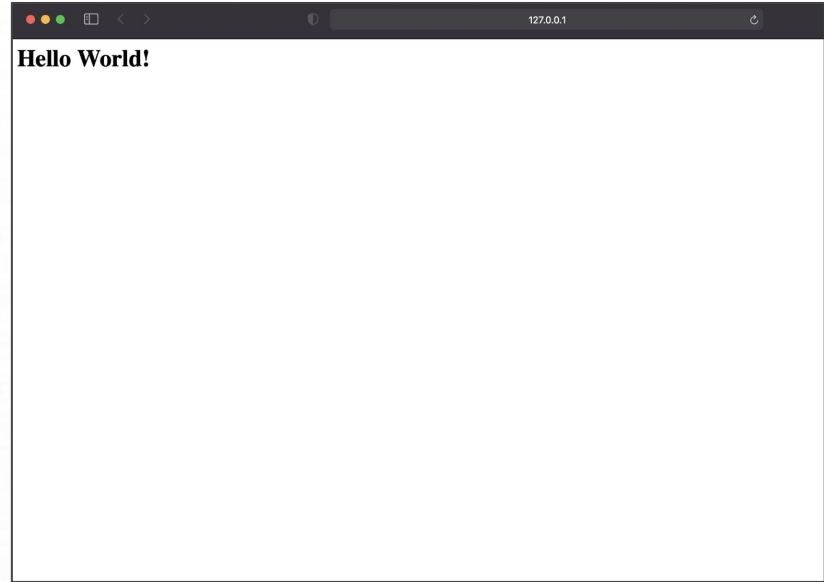
Language Features

Built-in Functions

- createHeader()
- createParagraph()
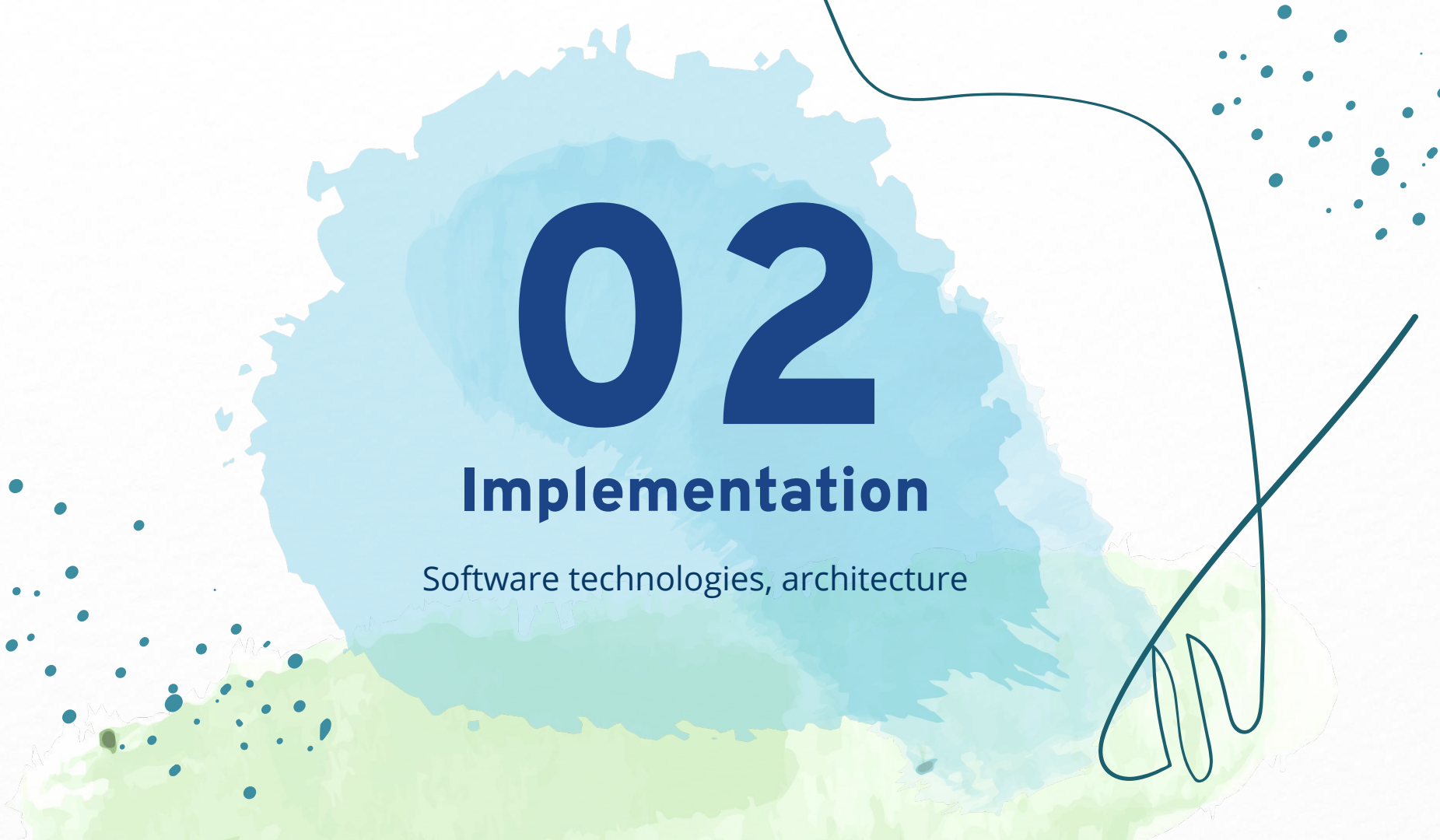- createSubheader()
- createList()
- createImage()

*modifications to print functions
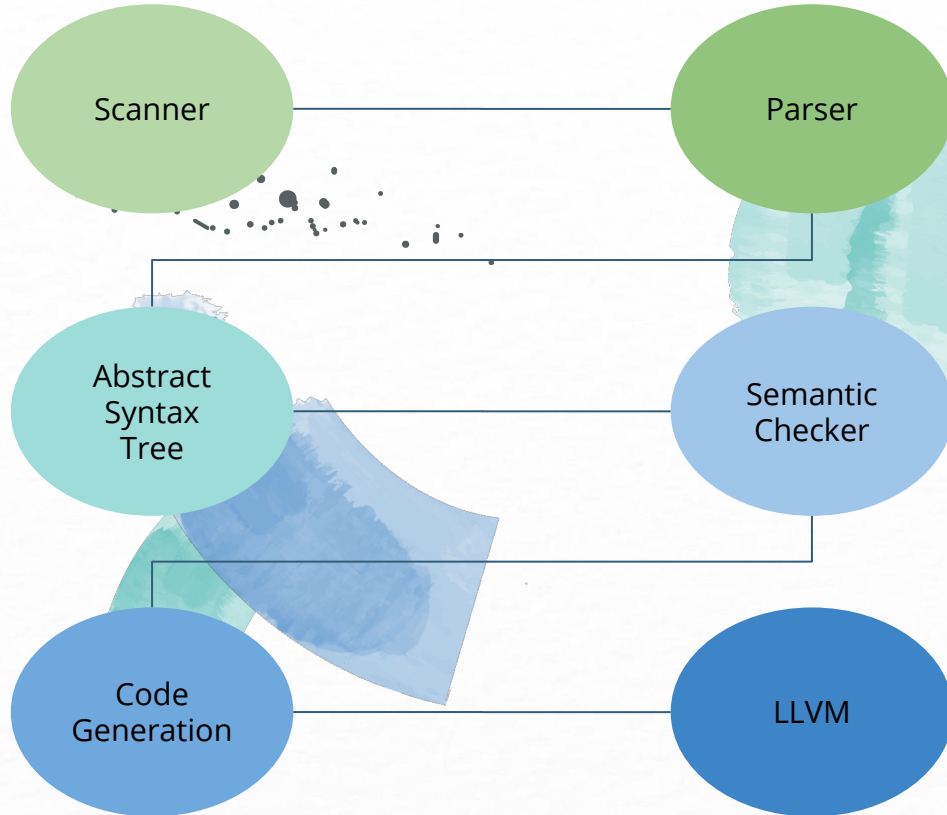
02

Implementation
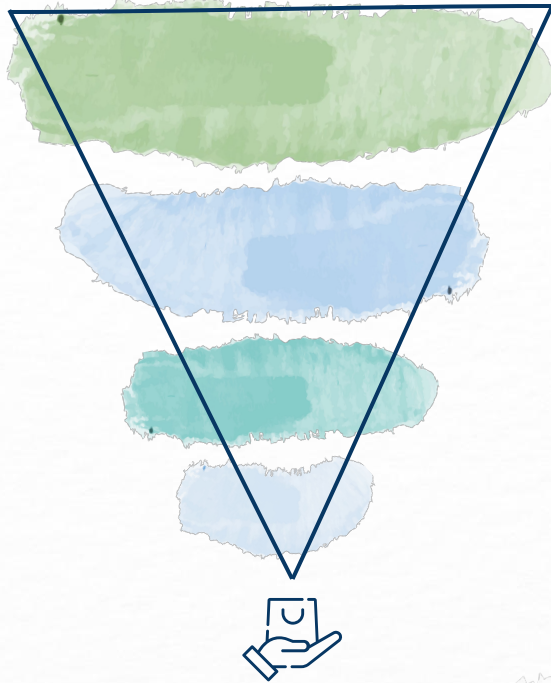
Software technologies, architecture

Architecture

03

Future Work

Additional features to implement

Future Work

01 **Data Types (Lists, Shapes, etc.)**

02 **Object-Oriented Programming**

03 **CSS Features**

04 **Pseudocode-Style Syntax**

Lessons Learned

Project Idea
Brainstorm early on and have a robust idea of plausible features to implement

Accountability
Make a project schedule with internal group deadlines and hold each other accountable

Office Hours
Attend office hours in addition to meetings with assigned TA

Milestones
Meet each of the class deadlines for project milestones; fix bugs sooner than later

Demo

THANK YOU!