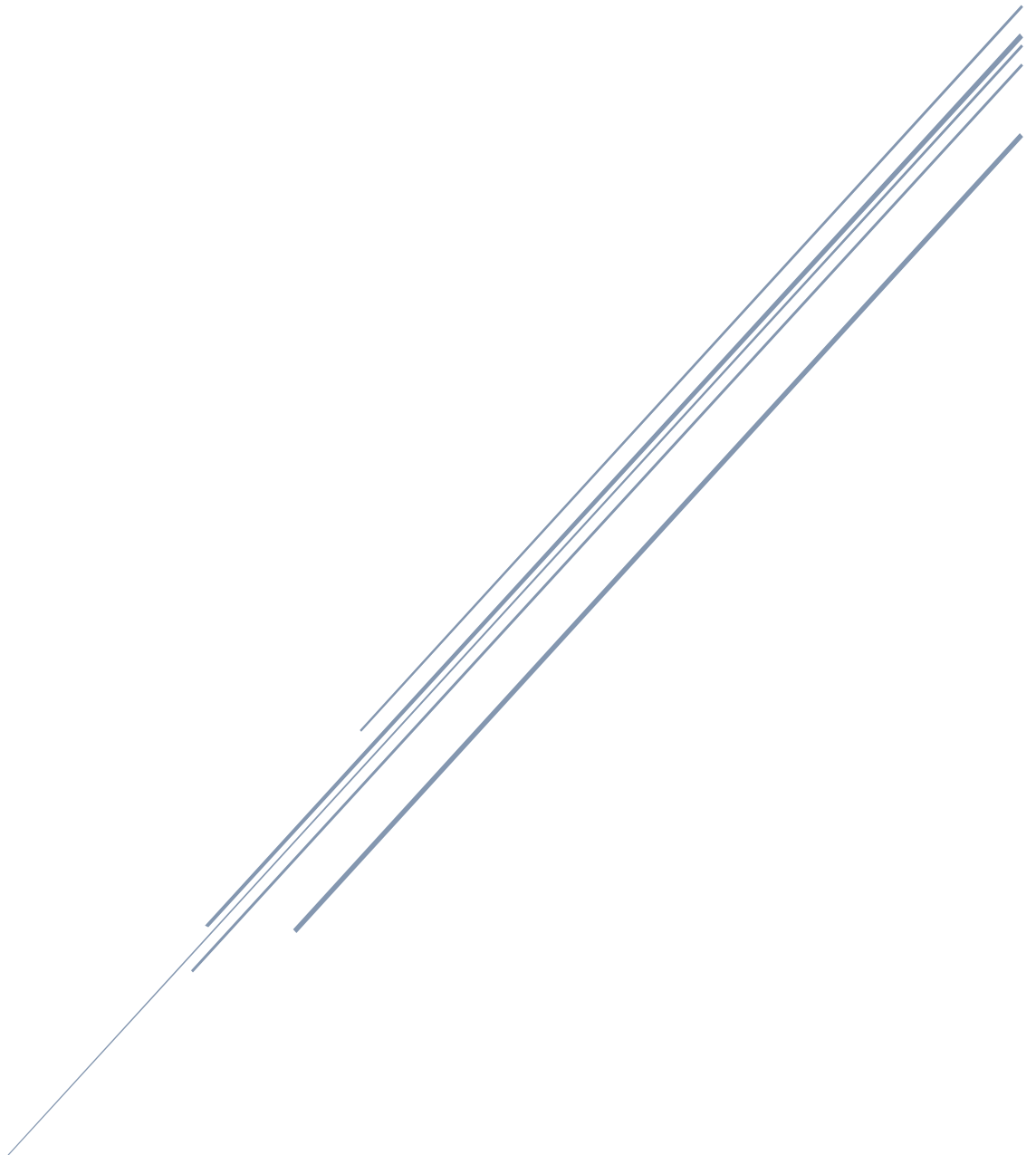


MSC SOFTWARE ENGINEERING (WEB DEVELOPMENT)

Web Services / Web Development



Jamie Ellis
P1001941

Figure 1- Backend example.....	5
Figure 2- Front end example.....	5
Figure 3- Statelessness frontend example.....	6
Figure 4- statelessness js example	6
Figure 5- Cache.....	7
Figure 6- Layer diagram	8
Figure 7- Repository code	8
Figure 8- Repository	9
Figure 9- Identification of resources	10
Figure 10- HATEOAS example	11
Figure 11- Diagram of Serverless AWS model [7]	13
Figure 12- Diagram of circuit-breaker pattern states [3].....	17

Table of Contents

Part A – Restful Web Services	4
What is a Restful Web Service?	4
The six constraints	4
Client-Server Architecture	5
Statelessness	6
Cacheability	7
Layered System	8
Code on demand	9
Uniform Interface	10
Part B– Serverless Technologies and	12
Circuit-breaker pattern.....	12
What is “Serverless”?	12
The Serverless Model	13
FaaS.....	14
Benefits and Drawbacks.....	15
Circuit breaker pattern	16
Closed	17
Open	17
Half-open	17
Analysis	18
Bibliography	19

Part A – Restful Web Services

This part of the report will look into Restful Web services and the best practises to put into play when implementing while showing examples from a web service I have been designing.

What is a Restful Web Service?

REST (Representational State Transfer) is the creation of Roy Fielding in his PHD Architectural Styles and the Design of Network-based Software Architectures. He best describes REST as being “an architectural style consisting of the set of constraints applied to elements within the architecture”. The idea of REST is to represent the best practices when designing a web service and below we will discuss the big 6 constraints that make up REST. [1]

The six constraints

The 6 constraints are:

1. Client-Server Architecture
2. Statelessness
3. Cacheability
4. Layered System
5. Code On Demand
6. Uniform Interface

[2]

Client-Server Architecture

This constraint looks to separate concerns between the client side- a front end for the client and the server side- backend service. It aims to take advantage of the powerful processors in today's web clients that are more common now unlike many years ago where these clients would be available to all computers- just powerful ones capable of running them. While also aiming to simplify the server side of things by shifting duties away from it to the client when appropriate. [3]

I have utilised a Client-Server Architecture in my design by designing a web service back end.

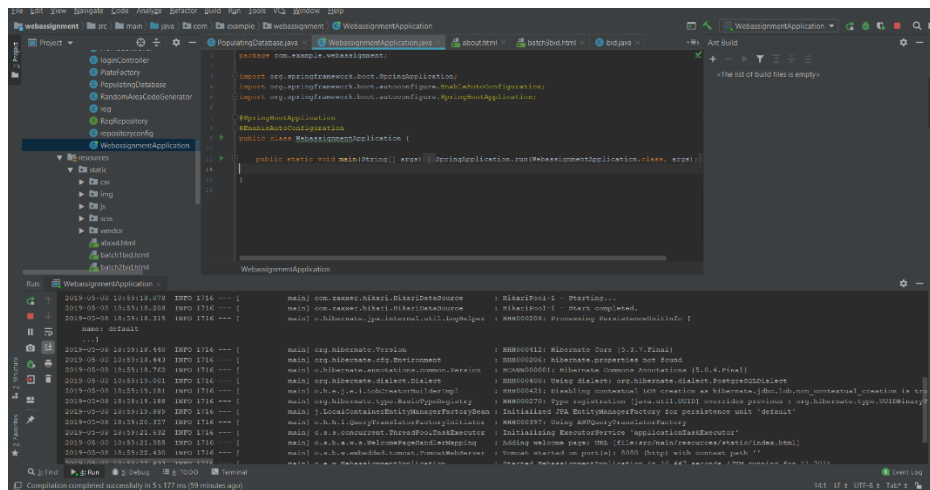


Figure 1- Backend example

I have implemented a front-end client web page service running off the web service as shown below.

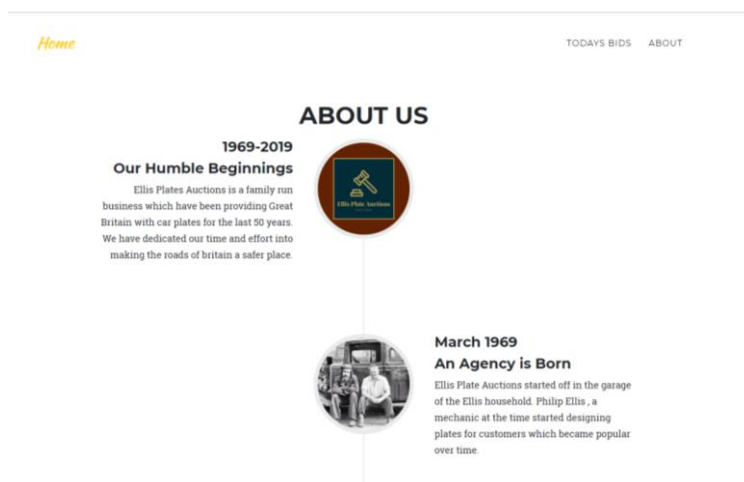
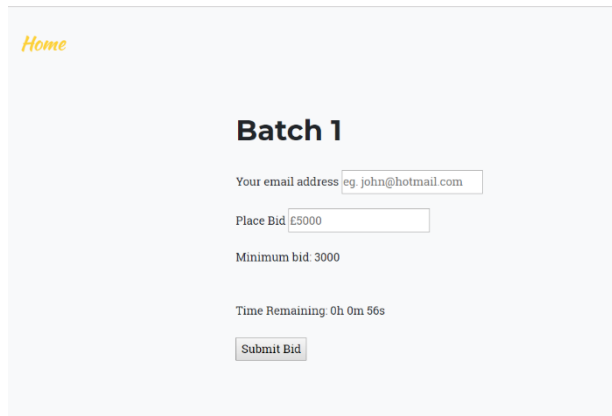


Figure 2- Front end example

Statelessness

One of Roy Fieldings constraints is the principle of Statelessness. This is taking into consideration when a client wants to make a request to the server. To achieve this, they will need to provide the necessary data to make the request. On the server-side though, the client must be provided with clear instructions so they can be clear on what the request is. Statelessness aims to achieve this and by doing so, it is said that the visibility, reliability and scalability of these requests can be improved using this constraint.[4] In my service, the client will create a request to the server when they wish to make a bid. They are provided with a form that provides clear instructions on the data they need to fill in and the request gets sent on when they click the submit button.



Home

Batch 1

Your email address

Place Bid

Minimum bid: 3000

Time Remaining: 0h 0m 56s

Figure 3- Statelessness frontend example

The server-side then reads the requests sent and communicates back if successful or not with the use of javascript and ajax.

```
function postbid() {  
    var biddata = JSON.stringify({  
        emailaddress: document.getElementById("username1").value,  
        bidamount: document.getElementById("highbid1").value  
    })  
    $.ajax({  
        url: "/bidding",  
        type: "POST",  
        dataType: "json",  
        contentType: "application/json",  
        data: biddata,  
        success: successbid,  
        error: errorbid,  
    })  
}  
function successbid(){  
    alert("Bid successful");  
}  
function errorbid(){  
    alert("Error = bid not submitted");  
}
```

Figure 4- statelessness js example

Visibility is improved as the service provided for looking after the requests will only need to do the one request to get the details they need to complete the request. [5] This is expanded on by John Flanders who says that "If a request contains everything needed for the server to make a proper reply, the request also contains all the data needed to track and report on that request". [4] Reliability gets improved as there is less worry about the failure of the requests. Scalability improves due to the server being able to process more requests as it is responsible. [5] It is backed up by John Flanders who claims that the manageability is also said to increase due to the state of a request being dealt with inside the request itself. [4]

Cacheability

A RESTful service can come across performance issues and one of the issues is caching. A service can have slow moments when trying to reload data repeatedly and REST deals with this problem in a clever way, data being stored away in memory can be reused to provide a boost to the performance of the user's side. It is a requirement of a REST service that data is declared whether it is meant for caching or not. [3]

As useful as it is, caching is also said to have some issues that need to be monitored. While it can give a performance boost, it can also store invalid data after some time if the data being saved isn't updating. To fix this, cache can be cleared to give more up-to-date data. It can be looked after in different ways:

- Client-side through an internet browser
- Server-side
- In the middle of both

[3]

I haven't implemented caching into my service for now due to parts of my system sending data and caching causing problems. When the system has been designed fully, I would look to implement and manage it successfully.

However, using Chrome on my service from the client saves caches of the data being sent into my service. This caused some problems testing and the caching had to be deleted on a regular basis as shown below.

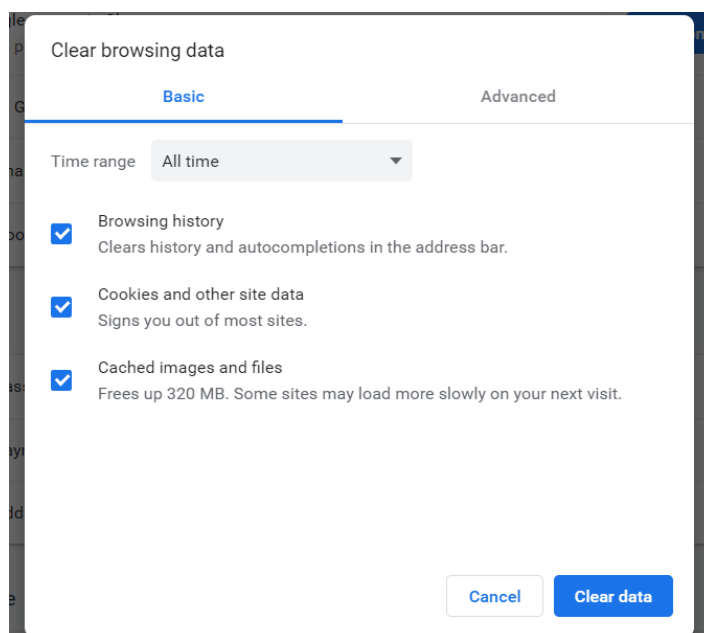


Figure 5- Cache

Layered System

REST was designed with the purpose of building a web service capable of dealing with excessive amounts of internet traffic and one of the ways to cope with this is through the use of a layered system. A layered system works by separating the system into layers by the components that make it up and allowing the layers to communicate through each other. [6]

The diagram below shows more clearly produced by Harihara Subramanian show a layered system communicates with each other. As you can see, the layers work their way down one layer at a time and can also work their way up but they must work their way to the bottom before heading back up the layers. [7]

The layers are flexible as in they can be reordered and modified throughout the creation of the service.

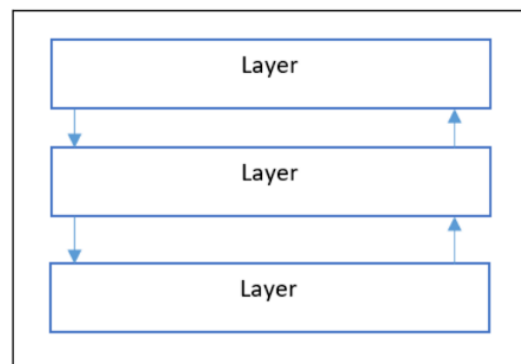


Figure 6- Layer diagram

This can have many benefits for example keeping it more simple to understand and when it comes to coupling. Having a less complex system is considered to be a benefit due to the complexity of a web service and with the amount of data passing through them it makes sense to keep your system as simple to understand as possible. [6]

In the REST web service that I have been designing, we have a generated plates java class that follows a layered system to achieve its purpose. It follows the flow of generating plates, talking to the batch and reg repositories using the line of code below. rr referring to the reg repository and then sending a POST request to the database.

```
//INSERT:  
pr.save(batchcount);  
  
for (int i = 0; i < c; i++) {
```

Figure 7- Repository code

To talk to the database, it has to go through the repository layer successfully and any new Post requests require the same process every time.

```
1 package com.example.webassignment;
2
3 import org.springframework.data.jpa.repository.Query;
4 import org.springframework.data.repository.PagingAndSortingRepository;
5 import org.springframework.data.rest.core.annotation.RepositoryRestResource;
6
7 import java.util.List;
8
9 @RepositoryRestResource(collectionResourceRel="batch", path="batch")
10 public interface BatchRepository extends PagingAndSortingRepository<batch, Integer> {
11
12 }
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

BatchRepository

Figure 8- Repository

Code on demand

The next constraint we will be looking at is code on demand. This constraint looks to give the client more functionality when using your service. This functionality can come in different forms like scripts for example and the purpose of them is to make the service to the client simpler by having pre-loaded scripts for them to use. [8] The clients can go into the service and download code that is pre-implemented by the server. [6]

This is not considered to be a constraint that is necessary to building a REST service, instead being offered as an optional constraint. The reason being for this is that it's said to reduce visibility in your system and careful consideration is needed before implementing it. [8]

Fernando Doglio gives his opinion on when he would use this constraint, he feels that a REST API does not require the need to implement the code on demand constraint as an API typically pulls information from endpoints and processes that information and it wouldn't be useful. However, he goes on to say that "for other uses of REST, like web servers, a client (i.e., a browser) will probably benefit from this constraint". [6]

I have decided not to use this constraint due to my service being primarily a REST API that relies on bringing information back from an endpoint. The use of scripts throughout the html pages were designed to bring back this information.

Uniform Interface

The final constraint that we will be discussing in this report will be Uniform Interface. This is considered an important constraint. Its concern is that data and messages being communicated to the client and server are shared. This makes the system simpler to maintain as it would be more of an issue if your system had different ways of communicating and having to get them talking to would more difficult than designing a system that communicates together. [9]

There are 4 constraints inside the Uniform Interface that enforces it:

- Identification of resources

The idea that each web service on your system will be known as a resource and have its own identifier such as a URL e.g. a login page could have /login.html to uniquely identify itself as a login page. [9]

Below is an example of implementing an identification of resources. Here I have made a path called /bids to represent the bids page.

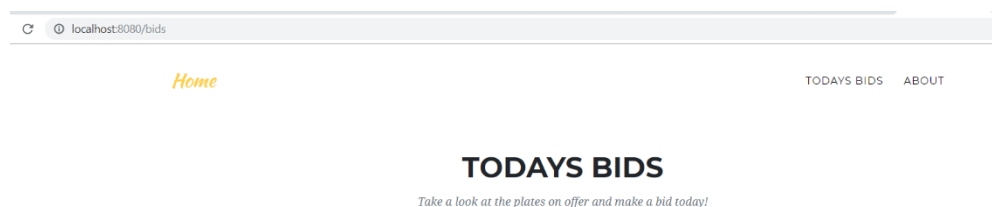


Figure 9- Identification of resources

- Manipulation of resources through representations

The representation of resources in different ways to the clients. Being able to represent the same data in different formats is important to this constraint for example representing customer information in the form of a HTML page but representing the same information through the use of JSON to pass it into a database. The data must never change but the way it is represented can change. [9]

- Self-descriptive messages

A message sent from the client's side to the server which a response has to be made from the server side to accept or deny the clients request. An example of this would be a client making a request to join a fantasy football league and the server side has to accept them into the league.[9]

- Hypermedia as the engine of application state (HATEOAS)

This constraint represents including links into your service to other services for the client to navigate through. This can be in the form of hyperlinks for example using a header for a webpage with a navigation bar to other pages. It can also be used to highlight important aspects of your service or just as a way for the client to navigate back to the homepage. [9]

Here's an example of the use of HATEOAS in the web service I have been designing. Here I have a home page with a navigation bar to the bids and about pages while also including a home link in the top left corner.

Home

TODAYS BIDS ABOUT



Welcome To Ellis Plate Auctions!

Figure 10- HATEOAS example

Part B– Serverless Technologies and

Circuit-breaker pattern

This report will take a look at “Serverless” technologies and their relevance to web hosting. This report will also be looking into the circuit-breaker pattern and how it can be useful in modern web development.

What is “Serverless”?

It has been described as the next part of the evolution away from the physical servers of the past. [9]

Despite the name "Serverless" , applications still require a server to run but the technology works by using an extra abstraction layer on top of the infrastructure placed away in the cloud. This gives the architectures less to worry about as the management and maintaining of the physical servers are replaced with servers that they access virtually and maintained elsewhere. [2]

Basically this leaves the developers with just the coding, testing and deploying left to worry about with everything else managed by the serverless provider. [8]

Serverless is not just a model, it can be used as an Idea and implemented in many different ways. It can be used to containerise applications as it provides the user with control over their containers that they're applications are running on. Where normally a developer would need to be running these systems, Serverless removes this by maintaining the systems and allowing the user to easily check up on the state of their containers. [1]

So let's take a look at the Serverless model.

The Serverless Model

The below diagram show all the different services that are provided by Amazon AWS in the Serverless model that they offer and I'll explain what each part of the model is providing.

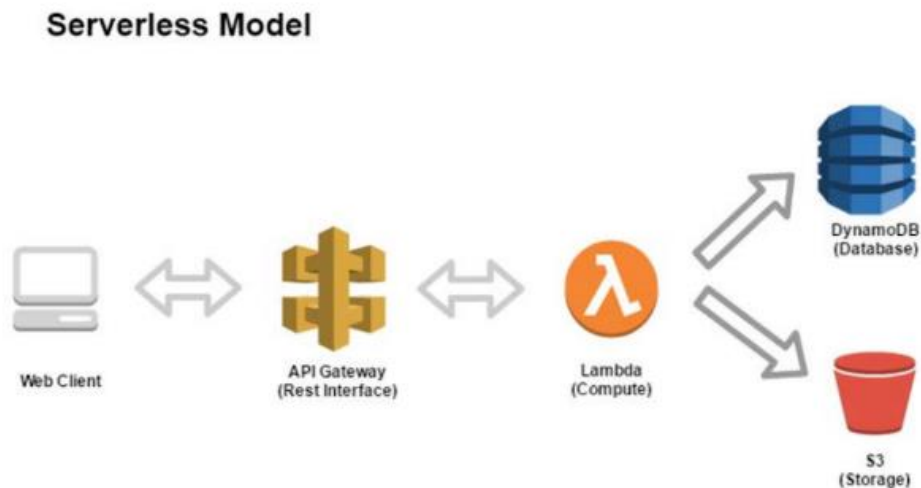


Figure 11- Diagram of Serverless AWS model [7]

API Gateway – The API gateway is used to create, maintain, monitor and deploy API services using a REST interface.

Lambda- This is where the application code gets executed.

DynamoDB – Once the code has been executed, data in the application gets stored into a DynamoDB database.

S3 Storage – An alternative to storing your data. In a S3 storage you can save your data in different formats. This can be used later on for example for data analytics. [7]

FaaS

FaaS (Function as a service) is the engine to make Serverless technology possible. It is described by Chanwit Kaewkasi as being “a computing engine, that allows us to write a simple, self-contained, single-purposed function to process or compute a task”. It is said to be acknowledged that a FaaS should be designed to be stateless as it allows the functions to be manageable. [5]

There are a variety of different ways that a FaaS platform can be implemented, not just in a Serverless environment. It can be run on your own hardware with the use of OpenFaaS and the Fn Project making this possible. But when implemented into the cloud, its known as a Serverless environment and called Serverless FaaS. [5]

There are different cloud companies which offer a Serverless environment:

- AWS - Lambda
- IBM - OpenWhisk
- Google Cloud Platform
- Microsoft Azure

Benefits and Drawbacks

Cost is considered to be a benefit when utilising Serverless technologies. The price of operational costs does not just affect the customers but also the cloud technology providers who offer Serverless. The providers see an increase in money due to the same infrastructure they provide being shared by everyone. The infrastructure set up is said to also help with the savings by not having applications running all day. The increase in money earned by the cloud providers gives them the opportunity to offer a cheaper service to the customer. They only need to pay for the services that they are running and the charges are not considered to be expensive, some services are free of charge. [2]

The ability to development at a much faster rate is another benefit that is said to come from Serverless technology. The majority of the infrastructure is managed by the cloud providers like Amazon AWS, Microsoft Azure etc. The developer is left with just writing code to apply to their business rules with an infrastructure already in place and this can save a lot of time to build and also the knowledge that the infrastructure is reliable. [2]

However, there are a few drawbacks to using Serverless for your infrastructure. The obvious one being that the servers running Serverless are not in your control and in some circumstances where more control is needed then Serverless would provide some limitations. Another drawback is that infrastructure designed personally makes it impossible to fully implement to a Serverless platform so in certain designs where a custom infrastructure would be required, then Serverless would not be possible. [5].

The possibility of being provided with a new set of tools to manage the testing, maintaining and deploying of your applications could mean that the tools that were being used before could be totally different and require staff to be trained on the new tools or maybe new staff with the skills to maintain the applications meaning the costs could become a problem despite the benefit of saving money from going to Serverless. [6]

The same can be considered to the security side of things. New security tools would normally be offered due to the difference of a Serverless environment and the change would need to adapted into the developers. It would make sense to consider these before making the decision to move over to Serverless. [6]

Circuit breaker pattern

The circuit breaker pattern was designed to solve the problem of waiting long times to get feedback on a problem with your application. It gets its name due to its similarities to an electricity circuit. According to Rhuan Rocha this is achieved with the pattern by preventing “an operation from repeatedly attempting to run in situations where it will likely fail during execution.” There is also a requirement in this pattern to run a check to see if the issue has been solved. [3]

This pattern can be very useful to modern web development due to the use of web services and the requests being made to the service. Cloud technology is gaining popularity and it is said that the possibility of a failure in the service still exists due to connection issues and services being unavailable to reach. Cloud technology would require some sort of fix to deal with these issues and the circuit-breaker pattern would offer a solution to many of the problems. [3]

So exactly how does it work?

The first thing you’d look at is the functions in your application and which ones are more likely to repeatedly fail. These functions can be fed into an object made for checking for failures. The pattern gets triggered when errors occur that they are searching for and the circuit breaker returns an error. [4]

The use of a proxy to link the request being made to the operation makes it a faster response as exceptions are returned immediately if the operation cannot be completed. There is said to be 3 different states that the proxy is said to have:

- Closed
- Open
- Half-open

[3].

The next part of the report will explain the difference in the states.

Closed

When a request is sent, the proxy is set in a closed state. Through the use of a counter which tracks the failures when an operation is executed it is able to know what state the proxy will be in. The counter gets incremented every time a failure occurs. The proxy will stay in a closed state until the counter hits a certain value that can be set by the developer. [3]

Open

The proxy gets changed to an open state when the failure counter limit is reached. When switched to an open state, a timer is set with a specific time set. The idea of this timer is to fix the failure in the time set. An exception request will be returned immediately in this process and once the timer has expired then the proxy state gets changed to the half-open state. [3]

Half-open

In this state, a number of changes are made to the pattern. Requests passed to the operation are limited, the failure counter is reset back to 0 and the state gets changed to closed. But any request failures force the state to reopen and the timer gets restarted making it not entirely closed or open. This state is said to have importance as it helps to reduce the requests and avoid having too many requests in a short space of time. [3]

The diagram below helps to understand more clearly the flow of the states and how they interact with each other. The diagram is a representation of the work of Rhuan Rocha. [3]

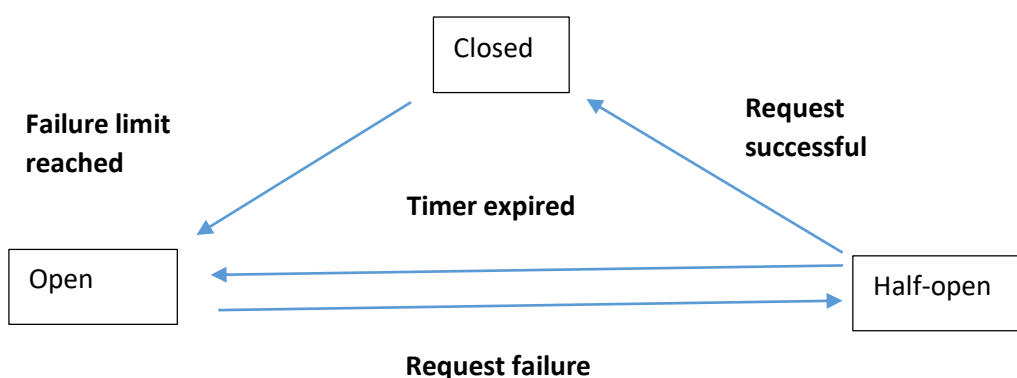


Figure 12- Diagram of circuit-breaker pattern states [3]

Analysis

The pattern is said to be crucial for:

- Monitoring
- Logging
- Recovering

[10]

These work towards ensuring that the service is available to use as much as possible.

It is also thought to help with the handling of downtime and slowness to services that have key importance by reducing the load to manage the service better. This is backed up by Martin Fowler who explains that circuit breakers help to reduce resources and goes on to explain why “You avoid waiting on timeouts for the client, and a broken circuit avoids putting load on a struggling server.”

[11]

Martin Fowler makes a point that although they are very important to include on your service, the developers using them have to react to failures when the circuit-breaker detects them. It is effectively a tool that can be used to highlight and deal with issues with your service and if both parts are done well then it can be used to many benefits. [11].

Bibliography

Part A

- [1] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," 2000.
- [2] "REST Principles and Architectural Constraints – REST API Tutorial," 2017. [Online]. Available: <https://restfulapi.net/rest-architectural-constraints/>. [Accessed: 08-May-2019].
- [3] C. Saternos, Client-server web apps with Javascript and Java.
- [4] J. Flanders, RESTful .NET. O'Reilly Media, 2008.
- [5] P. Klauzinski and J. Moore, Mastering JavaScript single page application development: an in-depth guide to exploring the design, architecture, and techniques behind building sophisticated, scalable, and maintainable single-page applications in JavaScript.
- [6] F. Doglio, Pro REST API Development with Node.js. Apress, 2015.
- [7] H. Subramanian and P. Raj, Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs. Packt Publishing, 2019.
- [8] C. Pautasso, E. Wilde, and R. Alarcon, REST: Advanced Research Topics and Practical Applications. Springer New York, 2013.
- [9] M. Massé, REST API design rulebook. O'Reilly, 2012.

Part B

- [1] M. Stigler, Beginning Serverless Computing: Developing with Amazon Web Services, Microsoft Azure, and Google Cloud. Apress, 2017.
- [2] K. Chowhan, Hands-On Serverless Computing: Build, run and orchestrate serverless applications using AWS Lambda, Microsoft Azure Functions, and Google Cloud Functions. Packt Publishing, 2018.
- [3] R. Rocha and J. Purificação, Java EE 8 Design Patterns and Best Practices: Build enterprise-ready scalable applications with architectural design patterns. Packt Publishing, 2018.
- [4] K. Ayevea and S. Kasampalis, Mastering Python Design Patterns: A guide to creating smart, efficient, and reusable software, 2nd Edition. Packt Publishing, 2018.
- [5] C. Kaewkasi, Docker for Serverless Applications: Containerize and orchestrate functions using OpenFaas, OpenWhisk, and Fn. Packt Publishing, 2018.
- [6] S. Rosenbaum, *Serverless computing in Azure with .NET*. Packt Publishing, 2017.
- [7] M. Gupta, Serverless Architectures with AWS: Discover how you can migrate from traditional deployments to serverless architectures with AWS. Packt Publishing, 2018.
- [8] D. Zanon, Building Serverless Web Applications. Packt Publishing, 2017.

[9] S. Burns, "SERVERLESS COMPUTING: WHAT IT IS AND WHY IT MATTERS: Enterprises might be forgiven for thinking serverless computing is just the latest in a long line of hyped technology concepts, but there is serious substance to it.," Comput. Wkly., pp. 21–23, 2018.

[10] D. Trivedi, "Circuit Breakers and Microservices Architecture | Constant Contact Tech Blog," 2017. [Online]. Available: <https://techblog.constantcontact.com/software-development/circuit-breakers-and-microservices/>. [Accessed: 05-May-2019].

[11] M. Fowler, "CircuitBreaker," 2014. [Online]. Available: <https://martinfowler.com/bliki/CircuitBreaker.html>. [Accessed: 05-May-2019].

Appendix - Web service

```
package com.example.webassignment;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class aboutController {
    @RequestMapping("/about")
    public String about()
    {
        return "about.html";
    }
}
```

```
package com.example.webassignment;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import javax.persistence.*;
import java.time.LocalDate;

@Entity
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class auction implements java.io.Serializable{
    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private int auctionid;
    private LocalDate auctiondate;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "bidid")
    private bid bidid;

    public auction()
    {
    }

    public auction(LocalDate auctiondate)
    {
        this.setAuctiondate(auctiondate);
    }

    public auction(String auctiondate)
    {
        this(LocalDate.parse(auctiondate));
    }

    public int getAuctionid() {
        return auctionid;
    }

    public void setAuctionid(int auctionid) {
        this.auctionid = auctionid;
    }

    public LocalDate getAuctiondate() {
        return auctiondate;
    }

    public void setAuctiondate(LocalDate auctiondate) {
        this.auctiondate = auctiondate;
    }

    public bid getBidid() {
        return bidid;
    }

    public void setBidid(bid bidid) {
        this.bidid = bidid;
    }
}

package com.example.webassignment;

import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
```

```

@RepositoryRestResource(collectionResourceRel="auction", path="auction")
public interface AuctionRepository extends PagingAndSortingRepository<auction, Integer> {

}

```

```

package com.example.webassignment;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import javax.persistence.*;
import java.util.List;

@Entity
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class batch implements java.io.Serializable {

    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private int batchid;

    private int minbatchprice;

    @OneToMany(fetch = FetchType.LAZY, targetEntity = reg.class, mappedBy = "batchid")
    private List<reg> regList;

    public batch() {

    }

    public batch(int batchid,int minbatchprice)

    {
        this.setBatchid(batchid);
        this.setMinbatchprice(minbatchprice);
    }

    public int getBatchid() {
        return batchid;
    }

    public void setBatchid(int batchid) {
        this.batchid = batchid;
    }

    public int getMinbatchprice() {
        return minbatchprice;
    }

    public void setMinbatchprice(int minbatchprice) {
        this.minbatchprice = minbatchprice;
    }

    public List<reg> getRegList() {
        return regList;
    }

    public void setRegList(List<reg> regList) {
        this.regList = regList;
    }

}

```

```

package com.example.webassignment;

import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import java.util.List;

@RepositoryRestResource(collectionResourceRel="batch", path="batch")
public interface BatchRepository extends PagingAndSortingRepository<batch, Integer> {

}

```

```

package com.example.webassignment;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import javax.persistence.*;
import java.util.List;

/**
 *
 * @author User-Admin
 */
@Entity
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class bid implements java.io.Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int bidid;
    private String emailaddress;
    private int bidamount;

    @OneToMany(fetch = FetchType.LAZY, targetEntity = auction.class, mappedBy = "bidid")
    private List<auction> auctionList;

    public bid() {
    }

    public bid(String emailaddress, int bidamount) {

        this.setEmailaddress(emailaddress);
        this.setBidamount(bidamount);
    }

    public int getBidid() {
        return bidid;
    }

    public void setBidid(int bidid) {
        this.bidid = bidid;
    }

    public String getEmailaddress() {
        return emailaddress;
    }

    public void setEmailaddress(String emailaddress) {
        this.emailaddress = emailaddress;
    }

    public int getBidamount() {
        return bidamount;
    }

    public void setBidamount(int bidamount) {
        this.bidamount = bidamount;
    }

    public List<auction> getAuctionList() {
        return auctionList;
    }

    public void setAuctionList(List<auction> auctionList) {
        this.auctionList = auctionList;
    }
}

```

```

package com.example.webassignment;

import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;

/**
 *
 * @author User-Admin
 */
@RepositoryRestResource(collectionResourceRel="bidding", path="bidding")
public interface BidRepository extends PagingAndSortingRepository<bid, Integer>
{

}

```

```

package com.example.webassignment;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class bidsController {
    @RequestMapping("/bids")
    public String bids()
    {
        return "todaysbids.html";
    }
}

```

```

package com.example.webassignment;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class contactController {
    @RequestMapping("/contact")
    public String contact()
    {
        return "contact.html";
    }
}

```

```

package com.example.webassignment;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HomeController {
    @RequestMapping("/home")
    public String home()
    {
        return "index.html"; /*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
}

package com.example.webassignment;

import java.util.Random;

/**
 *
 * @author User-Admin
 */
public class PlateFactory {

    public PlateFactory (){}

}

    public String getPlate(String alphabet, int alplength, int year, String areaCode) {

        Random RandomLetter = new Random();
        String finalLetters = "";
        for (int i = 0; i < 2; i++) {

            finalLetters += alphabet.charAt(RandomLetter.nextInt(alplength));

```

```

    }
    boolean badword = checkForRudeWords(finalLetters);
    do {
        {
            finalLetters += alphabet.charAt(RandomLetter.nextInt(alplength));
            checkForRudeWords(finalLetters);
        }
    }while (badword == true);

    int batchprice = getBatchPrice(finalLetters);

    String LicencePlate = areaCode + year + finalLetters;
    return LicencePlate;

}

public boolean checkForRudeWords(String finalLetters) {
    String[] rudeWordsList = {"Ass", "Tit", "fag", "Gay", "poo", "bum", "dic", "dik", "dck", "sck",
"sht", "vag"};
    boolean badword = false;
    for (int m = 0; m < rudeWordsList.length; m++) {
        if (finalLetters.contains(rudeWordsList[m])) {
            return true;
        }
        else
        {
            badword = false;
        }
    }
    return badword;
}

public int getBatchPrice(String finalLetters) {
    int batchPrice = 0;
    String[] keyWords = {"Tim", "Tom", "Jay", "Lee", "Amy", "Ric", "Mat", "Dan", "Dom", "Liz", "Rob",
"Kie"};
    for (int q = 0; q < keyWords.length; q++) {
        if (finalLetters.contains(keyWords[q])) {
            // batchPrice = 500;
            return 500;
        }
        else
        {
            batchPrice = 300;
        }
    }

    return batchPrice;
}

}

}

}

```

```

package com.example.webassignment;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import javax.annotation.PostConstruct;

@Service
public class PopulatingDatabase
{
    private final RegRepository rr;
    private final BatchRepository br;

    @Autowired
    public PopulatingDatabase(RegRepository rr, BatchRepository br)
    {
        this.rr = rr;
        this.br = br;
    }

    @PostConstruct
    @Transactional
    public void fillData()
    {
        //clear reg and batch table

        rr.deleteAll();
    }
}

```

```

//GENERATING PLATES:
String alphabet = "ABCDEFGHGIJKLMNOPQRSTUVWXYZ";
int N = alphabet.length();
int year = 19;

int d = 3;

for (int k = 1; k <=d; k++)
{
    RandomAreaCodeGenerator ra = new RandomAreaCodeGenerator();
    String areaCode = ra.getAreaCode(alphabet,N,year);

    batch batchcount = new batch();
    batchcount.setBatchid(k);
    int batchmin = 0;
    batchcount.setMinbatchprice(batchmin);

    int c = 10;

    //INSERT:
    br.save(batchcount);

    for (int i = 0; i <c; i++){
        PlateFactory pf = new PlateFactory();
        String LicencePlate = pf.getPlate(alphabet,N,year,areaCode);
        rr.save(new reg(LicencePlate, batchcount));
        batchmin += pf.getBatchPrice(LicencePlate);
    }
    //INSERT WITH BATCHPRICE
    batchcount.setMinbatchprice(batchmin);
    br.save(batchcount);
}
}
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.example.webassignment;

import java.util.Random;

/**
 *
 * @author User-Admin
 */
public class PlateFactory {

    public PlateFactory (){}

    public String getPlate(String alphabet, int alplength, int year, String areaCode) {

        Random RandomLetter = new Random();
        String finalLetters = "";
        for (int i = 0; i < 2; i++) {

            finalLetters += alphabet.charAt(RandomLetter.nextInt(alplength));

        }
        boolean badword = checkForRudeWords(finalLetters);
        do {
            {
                finalLetters += alphabet.charAt(RandomLetter.nextInt(alplength));
                checkForRudeWords(finalLetters);
            }
        }while (badword == true);

        int batchprice = getBatchPrice(finalLetters);

        String LicencePlate = areaCode + year + finalLetters;
        return LicencePlate;

    }
    public boolean checkForRudeWords(String finalLetters) {
        String[] rudeWordsList = {"Ass", "Tit", "fag", "Gay", "poo", "bum", "dic", "dik", "dck", "sck",
        "sht", "vag"};
        boolean badword = false;
    }
}

```

```

        for (int m = 0; m < rudeWordsList.length; m++) {
            if (finalLetters.contains(rudeWordsList[m])) {
                return true;
            }
            else
            {
                badword = false;
            }
        }
        return badword;
    }

    public int getBatchPrice(String finalLetters) {
        int batchPrice = 0;
        String[] keyWords = {"Tim", "Tom", "Jay", "Lee", "Amy", "Ric", "Mat", "Dan", "Dom", "Liz", "Rob",
"Kie"};
        for (int q = 0; q < keyWords.length; q++) {
            if (finalLetters.contains(keyWords[q])) {
                // batchPrice = 500;
                return 500;
            }
            else
            {
                batchPrice = 300;
            }
        }

        return batchPrice;
    }
}

package com.example.webassignment;

import java.util.Random;

public class RandomAreaCodeGenerator {

    public RandomAreaCodeGenerator () {

    }

    public String getAreaCode (String alphabet,int alplength, int year)
    {

        Random RandomLetter = new Random();
        String acLetters = "";
        for (int i = 0; i < 2; i++)
        {
            acLetters += alphabet.charAt(RandomLetter.nextInt(alplength));
        }

        String areaCode = acLetters;
        return areaCode;
    }
}

```

```

package com.example.webassignment;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import javax.persistence.*;

@Entity
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class reg implements java.io.Serializable{
    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private int regid;
    private String regplate;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "batchid")
    private batch batchid;

    public reg()
    {
    }

    public reg(String regplate)
    {
        this.setRegplate(regplate);
    }

    public reg(String regplate,batch batchid)
    {
    }
}

```

```

        this.setRegplate(regplate);
        this.setBatchid(batchid);
    }

//    public reg(String regplate,int batchid)
//    {
//        this.setRegplate(regplate);
//        this.setBatchid(batchid);
//    }

    public int getRegid() {
        return regid;
    }

    public void setRegid(int regid) {
        this.regid = regid;
    }

    public String getRegplate() {
        return regplate;
    }

    public void setRegplate(String regplate) {
        this.regplate = regplate;
    }

    public batch getBatchid() {
        return batchid;
    }

    public void setBatchid(batch batchid) {
        this.batchid = batchid;
    }
}

```

```

package com.example.webassignment;

import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource(collectionResourceRel="reg", path="reg")
public interface RegRepository extends PagingAndSortingRepository<reg, Integer> {
}

```

```

package com.example.webassignment;

import org.springframework.context.annotation.Configuration;
import org.springframework.data.rest.core.config.RepositoryRestConfiguration;
import org.springframework.data.rest.webmvc.config.RepositoryRestConfigurerAdapter;

@Configuration
public class repositoryconfig extends RepositoryRestConfigurerAdapter {
    @Override
    public void configureRepositoryRestConfiguration(RepositoryRestConfiguration config) {
        config.exposeIdsFor(reg.class);
        config.exposeIdsFor(batch.class);
        config.exposeIdsFor(bid.class);
        config.exposeIdsFor(auction.class);
    }
}

```

```

package com.example.webassignment;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@EnableAutoConfiguration
public class WebassignmentApplication {

    public static void main(String[] args) {
        SpringApplication.run(WebassignmentApplication.class, args);
    }
}

```



```

                <h4>March 1969</h4>
                <h4 class="subheading">An Agency is Born</h4>
            </div>
            <div class="timeline-body">
                <p class="p2">Ellis Plate Auctions started off in the garage of the
Ellis household. Philip Ellis , a mechanic at the time started designing plates for customers which became
popular over time. </p>
            </div>
        </li>
        <li>
            <div class="timeline-image">
                
            </div>
            <div class="timeline-panel">
                <div class="timeline-heading">
                    <h4>December 2012</h4>
                    <h4 class="subheading">Government Partnership</h4>
                </div>
                <div class="timeline-body">
                    <p class="p3">Under the leadership of his son , Jamie. Ellis Plate
Auctions were able to form a partnership with the civil service to produce licence plates for customers
across the UK becoming the leading company in auctioning plates.</p>
                </div>
            </div>
            <li class="timeline-inverted">
                <div class="timeline-image">
                    <h4>Be Part
                    <br>Of Our
                    <br>Story!</h4>
                </div>
            </li>
        </ul>
    </div>
</div>
</section>

</body>

</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- use of bootstrap with design -->
    <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"
type="text/css">
    <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet'
type='text/css">
    <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css">
    <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'
type='text/css">
    <link href="css/ellisplateauctions.css" rel="stylesheet">

</head>
<body id="page-top">

    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
        <div class="container">
            <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
            <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">

                Menu
                <i class="fas fa-bars"></i>
            </button>
            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav text-uppercase ml-auto">

                    <li class="nav-item">
                        <a class="nav-link js-scroll-trigger" href= "todaysbids.html">Todays Bids</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link js-scroll-trigger" href="about.html">About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

```

```

        </ul>
      </div>
    </div>
  </nav>
  <section class="bg-light" id="login">
    <div id="bidform" >
      <form>

        <h1>Batch 1 </h1>
        <br>
        <p>
          <label for="username" class="uname" data-icon="u"> Your email address
        </label>
          <input id="username1" name="username" required="required" type="text"
placeholder="eg. john@hotmail.com"/>
        </p>
        <p>
          <label for="bidvalue" class="bvalue" data-icon="bv" > Place Bid
        </label>
          <input id="bidvalue1" name="bidvalue" required="required" type="number"
placeholder="£5000"/>
        </p>
        <p>
          <label for="minbid1" class="mbid" data-icon="mb" id = "minbid1">
        </label>
          <p>
            <label for="highestbid1" class="hbid" data-icon="mb" id = "highbid1"
value=0></label>
            <p>
              <label for="timer" class="timer"
data-icon="time1" id="timer" > Time Remaining: </label>
              <label for="timer1" class="timer1" data-icon="time1" id="timer1" > </label>
            </p>
            <p class="bid button">
              <button id="submit1" type="button">Submit Bid</button>
            </p>
          </p>
        </form>
      </div>
    </section>

    <script src="vendor/jquery/jquery.min.js"></script>
    <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
    <script src="js/batchl auction.js"></script>
    <script src="js/minbid.js"></script>

  </body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <!-- use of bootstrap with design -->
  <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"
type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet'
type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'
type='text/css'>
  <link href="css/ellisplateauctions.css" rel="stylesheet">

</head>
<body id="page-top">

  <!-- Navigation -->
  <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
    <div class="container">
      <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
      <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">
        Menu

```

```

        <i class="fas fa-bars"></i>
      </button>
      <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav text-uppercase ml-auto">
          <li class="nav-item">
            <a class="nav-link js-scroll-trigger" href= "todaysbids.html">Todays Bids</a>
          </li>
          <li class="nav-item">
            <a class="nav-link js-scroll-trigger" href="about.html">About</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <section class="bg-light" id="login">
    <div id="bidform" >
      <form>

        <h1>Batch 2 </h1>
        <br>
        <p>
          <label for="username" class="uname" data-icon="u"> Your email address
        </label>
          <input id="username2" name="username" required="required" type="text"
placeholder="eg. john@hotmail.com"/>
        </p>
        <p>
          <label for="bidvalue" class="bvalue" data-icon="bv" > Place Bid
        </label>
          <input id="bidvalue2" name="bidvalue" required="required" type="number"
placeholder="£5000"/>
        </p>
        <p>
          <label for="minbid2" class="mbid" data-icon="mb" id = "minbid2">
        </label>
          <p>
            <label for="highestbid2" class="hbid" data-icon="mb" id = "highbid2"
value=0></label>
          <p>
            <label for="timer2" class="timer2" data-icon="time2" id="timer" > Time
Remaining: </label>
            <label for="timer2" class="timer2" data-icon="time2" id="timer2" > </label>
          </p>
          <p class="bid button">
            <button id="submit2" type="button">Submit Bid</button>
          </p>
        </form>
      </div>
    </section>

<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="js/batch2auction.js"></script>
<script src="js/minbid2.js"></script>

</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <!-- use of bootstrap with design -->
  <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"
type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet'
type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'
type='text/css'>
  <link href="css/ellisplateauctions.css" rel="stylesheet">

</head>
<body id="page-top">

```

```

<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
  <div class="container">
    <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
    <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">
      Menu
      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <ul class="navbar-nav text-uppercase ml-auto">

        <li class="nav-item">
          <a class="nav-link js-scroll-trigger" href= "todaysbids.html">Todays Bids</a>
        </li>
        <li class="nav-item">
          <a class="nav-link js-scroll-trigger" href="about.html">About</a>
        </li>

      </ul>
    </div>
  </div>
</nav>
<section class="bg-light" id="login">
  <div id="bidform" >
    <form>

      <h1>Batch 3 </h1>
      <br>
      <p>
        <label for="username" class="uname" data-icon="u"> Your email address
      </label>
        <input id="username3" name="username" required="required" type="text"
placeholder="eg. john@hotmail.com"/>
      </p>
      <p>
        <label for="bidvalue" class="bvalue" data-icon="bv" > Place Bid
      </label>
        <input id="bidvalue3" name="bidvalue" required="required" type="number"
placeholder="£5000"/>
      </p>
      <p>
        <label for="minbid3" class="mbid" data-icon="mb" id = "minbid3">
      </label>
      <p>
        <label for="highestbid3" class="hbid" data-icon="mb" id = "highbid3"
value=0></label>
      <p>
        <label for="timer3" class="timer3" data-icon="time3" id="timer" > Time
Remaining: </label>
        <label for="timer3" class="timer3" data-icon="time3" id="timer3" > </label>
      </p>
      <p class="bid button">
        <button id="submit3" type="button">Submit Bid</button>
      </p>
    </form>
  </div>
</section>

<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="js/batch3auction.js"></script>
<script src="js/minbid3.js"></script>

</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">
<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Ellis Plate Auctions</title>

  <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"

```

```

type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'
type='text/css'>
  <link href="css/ellisplateauctions.css" rel="stylesheet">

</head>

<body id="page-top">

  <!-- Navigation -->
  <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
    <div class="container">
      <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
      <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">
        Menu
        <i class="fas fa-bars"></i>
      </button>
      <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav text-uppercase ml-auto">
          <li class="nav-item">
            <a class="nav-link js-scroll-trigger" href="todaysbids.html">Todays Bids</a>
          </li>
          <li class="nav-item">
            <a class="nav-link js-scroll-trigger" href="about.html">About</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <!-- Header -->
  <header class="masthead">
    <div class="container">
      <div class="intro-text">

        
        <br>
        <br>
        <div class="intro-lead-in">Welcome To Ellis Plate Auctions!</div>
        <a class="btn btn-primary btn-xl text-uppercase js-scroll-trigger"
href="todaysbids.html">Auctions</a>
      </div>
    </div>
  </header>

  <!-- Bootstrap core JavaScript -->
  <script src="vendor/jquery/jquery.min.js"></script>
  <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

</body>
</html>

```

```

<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"
type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet'
type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'
type='text/css'>
  <link href="css/agency.css" rel="stylesheet">

</head>

<body id="page-top">

  <!-- Navigation -->
  <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
    <div class="container">
      <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
      <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">

```

```

        Menu
        <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav text-uppercase ml-auto">
            <li class="nav-item">
                <a class="nav-link js-scroll-trigger" href="login.html">Login/Regisiter</a>
            </li>
            <li class="nav-item">
                <a class="nav-link js-scroll-trigger" href="todaysbids.html">Todays Bids</a>
            </li>
            <li class="nav-item">
                <a class="nav-link js-scroll-trigger" href="about.html">About</a>
            </li>
            <li class="nav-item">
                <a class="nav-link js-scroll-trigger" href="contact.html">Contact</a>
            </li>
        </ul>
    </div>
</div>
</nav>
<section class="bg-light" id="register">
    <div id="registerForm">
        <form>
            <h1> Sign up </h1>
            <br>
            <p>
                <label for="username" class="uname" data-icon="u">Your
username</label>
                <input id="username" name="username" required="required"
type="text" placeholder="mysuperusername690" />
            </p>
            <p>
                <label for="email" class="youmail" data-icon="e" > Your
email</label>
                <input id="email" name="email" required="required"
type="email" placeholder="mysupermail@mail.com" />
            </p>
            <p>
                <label for="password" class="youpasswd" data-icon="p">Your
password </label>
                <input id="password" name="password" required="required"
type="password" placeholder="eg. X8df!90EO" />
            </p>
            <p>
                <label for="passwordconfirm" class="youpasswd" data-
icon="p">Please confirm your password </label>
                <input id="passwordconfirm" name="passwordconfirm"
required="required" type="password" placeholder="eg. X8df!90EO" />
            </p>
            <p class="signin button">
                <input type="submit" value="Sign up" />
            </p>
            <p class="change link">
                Already a member ?
                <a href="login.html" class="to_register"> Go and log in </a>
            </p>
        </form>
    </div>

    <script src="vendor/jquery/jquery.min.js"></script>
    <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- use of bootstrap with design -->
    <link href="vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet"
type="text/css">
    <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet'

```

```

type='text/css'>
    <link href="css/ellisplateauctions.css" rel="stylesheet">

</head>

<body id="page-top">

    <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
        <div class="container">
            <a class="navbar-brand js-scroll-trigger" href="index.html">Home</a>
            <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">
                Menu
                <i class="fas fa-bars"></i>
            </button>
            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav text-uppercase ml-auto">

                    <li class="nav-item">
                        <a class="nav-link js-scroll-trigger" href="todaysbids.html">Todays Bids</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link js-scroll-trigger" href="about.html">About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <section id="todaysbids">
        <div class="container">
            <div class="row">
                <div class="col-lg-12 text-center">
                    <h2 class="section-heading text-uppercase">Todays Bids</h2>
                    <h3 class="section-subheading text-muted">Take a look at the plates on offer and make a
bid today!</h3>
                </div>
            </div>
            <div class="row text-center">
                <div class="col-md-4">
                    <span class="fa-stack fa-4x">
                        <i class="fas fa-circle fa-stack-2x text-primary"></i>
                        <i class="fas fa-shopping-cart fa-stack-1x fa-inverse"></i>
                    </span>
                    <button id="batch1" type="button">Batch 1</button>
                    <br>
                    <div class="container2">
                        <div class="row">
                            <div class="col-sm-4"></div>
                            <span></span>
                            <p class="text-muted1">Click button to see plates</p>
                            <a class="btn btn-primary btn-xl text-uppercase js-scroll-trigger"
href="batch1bid.html">Bid</a>
                        </div>
                    </div>
                </div>
                <div class="col-md-4">
                    <span class="fa-stack fa-4x">
                        <i class="fas fa-circle fa-stack-2x text-primary"></i>
                        <i class="fas fa-shopping-cart fa-stack-1x fa-inverse"></i>
                    </span>
                    <button id="batch2" type="button">Batch 2</button>
                    <br>
                    <div class="container2">
                        <div class="row">
                            <div class="col-sm-4"></div>
                            <p class="text-muted2">Click button to see plates</p>
                            <a class="btn btn-primary btn-xl text-uppercase js-scroll-trigger"
href="batch2bid.html">Bid</a>
                        </div>
                    </div>
                </div>
                <div class="col-md-4">
                    <span class="fa-stack fa-4x">
                        <i class="fas fa-circle fa-stack-2x text-primary"></i>
                        <i class="fas fa-shopping-cart fa-stack-1x fa-inverse"></i>
                    </span>
                    <button id="batch3" type="button">Batch 3</button>
                    <br>
                    <div class="container2">
                        <div class="row">
                            <div class="col-sm-4"></div>
                            <p class="text-muted3">Click button to see plates</p>
                            <a class="btn btn-primary btn-xl text-uppercase js-scroll-trigger"
href="batch3bid.html">Bid</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>

```

```

        </div>
      </div>
    </div>
  </div>
  <script src="vendor/jquery/jquery.min.js"></script>
  <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="js/regplates.js"></script>

</section>

</body>
</html>

```

CSS

```

/*parts of css suppoorted by bootstrap*/

body {
  overflow-x: hidden;
  font-family: 'Roboto Slab', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

p {
  line-height: 1.75;
}

a {
  color: #fed136;
}

a:hover {
  color: #fec503;
}

.text-primary {
  color: #fed136 !important;
}

h1,
h2,
h3,
h4,
h5,
h6 {
  font-weight: 700;
  font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

section {
  padding: 100px 0;
}

HomePic {

  position: absolute;
  left: 0px;
  top: 0px;
}

section h2.section-heading {
  font-size: 40px;
  margin-top: 0;
  margin-bottom: 15px;
}

section h3.section-subheading {
  font-size: 16px;
  font-weight: 400;
  font-style: italic;
  margin-bottom: 75px;
  text-transform: none;
  font-family: 'Droid Serif', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

@media (min-width: 768px) {
  section {
    padding: 150px 0;
  }
}

```

```

.btn {
  font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
  font-weight: 700;
}

.btn-xl {
  font-size: 18px;
  padding: 20px 40px;
  text-align: center;
  justify-content: center;
}

.btn-primary {
  background-color: #fed136;
  border-color: #fed136;
}

.btn-primary:active, .btn-primary:focus, .btn-primary:hover {
  background-color: #fec810 !important;
  border-color: #fec810 !important;
  color: white;
}

.btn-primary:active, .btn-primary:focus {
  -webkit-box-shadow: 0 0 0 0.2rem rgba(254, 209, 55, 0.5) !important;
  box-shadow: 0 0 0 0.2rem rgba(254, 209, 55, 0.5) !important;
}

::selection {
  background: #fed136;
  text-shadow: none;
}

#mainNav .navbar-toggler {
  font-size: 12px;
  right: 0;
  padding: 13px;
  text-transform: uppercase;
  color: #fec503;
  border: 0;
  background-color: #fed136;
  font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

#mainNav .navbar-brand {
  color: #fed136;
  font-family: 'Kaushan Script', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

#mainNav .navbar-brand.active, #mainNav .navbar-brand:active, #mainNav .navbar-brand:focus, #mainNav
.navbar-brand:hover {
  color: #fec503;
}

#mainNav .navbar-nav .nav-item .nav-link {
  font-size: 90%;
  font-weight: 400;
  padding: 0.75em 0;
  letter-spacing: 1px;
  color: black;
  font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
  Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

#mainNav .navbar-nav .nav-item .nav-link.active, #mainNav .navbar-nav .nav-item .nav-link:hover {
  color: #fed136;
}

@media (min-width: 992px) {
  #mainNav {
    padding-top: 25px;
    padding-bottom: 25px;
    -webkit-transition: padding-top 0.3s, padding-bottom 0.3s;
    transition: padding-top 0.3s, padding-bottom 0.3s;
    border: none;
    background-color: transparent;
  }
  #mainNav .navbar-brand {
    font-size: 1.75em;
    -webkit-transition: all 0.3s;
    transition: all 0.3s;
  }
  #mainNav .navbar-nav .nav-item .nav-link {
    padding: 1.1em 1em !important;
  }
  #mainNav .navbar-shrink {
    padding-top: 0;
  }
}

```

```

        padding-bottom: 0;
        background-color: #212529;
    }
    #mainNav.navbar-shrink .navbar-brand {
        font-size: 1.25em;
        padding: 12px 0;
    }
}

header.masthead {
    text-align: center;
    color: black;
    background-repeat: no-repeat;
    background-attachment: scroll;
    background-position: center center;
    background-size: cover;
}

header.masthead .intro-text {
    padding-top: 150px;
    padding-bottom: 100px;
}

header.masthead .intro-text .intro-lead-in {
    font-size: 22px;
    font-style: italic;
    line-height: 22px;
    margin-bottom: 25px;
    font-family: 'Droid Serif', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
    Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

header.masthead .intro-text .intro-heading {
    font-size: 50px;
    font-weight: 700;
    line-height: 50px;
    margin-bottom: 25px;
    font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
    Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
}

@media (min-width: 768px) {
    header.masthead .intro-text {
        padding-top: 300px;
        padding-bottom: 200px;
    }
    header.masthead .intro-text .intro-lead-in {
        font-size: 40px;
        font-style: italic;
        line-height: 40px;
        margin-bottom: 25px;
        font-family: 'Droid Serif', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
        Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
    }
    header.masthead .intro-text .intro-heading {
        font-size: 75px;
        font-weight: 700;
        line-height: 75px;
        margin-bottom: 50px;
        font-family: 'Montserrat', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue',
        Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
    }
}

.service-heading {
    margin: 15px 0;
    text-transform: none;
}

.timeline {
    position: relative;
    padding: 0;
    list-style: none;
}

.timeline:before {
    position: absolute;
    top: 0;
    bottom: 0;
    left: 40px;
    width: 2px;
    margin-left: -1.5px;
    content: '';
    background-color: #e9ecef;
}

.timeline > li {
    position: relative;
    min-height: 50px;

```

```

margin-bottom: 50px;
}

.timeline > li:after, .timeline > li:before {
display: table;
content: ' ';
}

.timeline > li:after {
clear: both;
}

.timeline > li .timeline-panel {
position: relative;
float: right;
width: 100%;
padding: 0 20px 0 100px;
text-align: left;
}

.timeline > li .timeline-panel:before {
right: auto;
left: -15px;
border-right-width: 15px;
border-left-width: 0;
}

.timeline > li .timeline-panel:after {
right: auto;
left: -14px;
border-right-width: 14px;
border-left-width: 0;
}

.timeline > li .timeline-image {
position: absolute;
z-index: 100;
left: 0;
width: 80px;
height: 80px;
margin-left: 0;
text-align: center;
color: white;
border: 7px solid #e9ecef;
border-radius: 100%;
background-color: #fed136;
}

.timeline > li .timeline-image h4 {
font-size: 10px;
line-height: 14px;
margin-top: 12px;
}

.timeline > li.timeline-inverted > .timeline-panel {
float: right;
padding: 0 20px 0 100px;
text-align: left;
}

.timeline > li.timeline-inverted > .timeline-panel:before {
right: auto;
left: -15px;
border-right-width: 15px;
border-left-width: 0;
}

.timeline > li.timeline-inverted > .timeline-panel:after {
right: auto;
left: -14px;
border-right-width: 14px;
border-left-width: 0;
}

.timeline > li:last-child {
margin-bottom: 0;
}

.timeline .timeline-heading h4 {
margin-top: 0;
color: inherit;
}

.timeline .timeline-heading h4.subheading {
text-transform: none;
}

.timeline .timeline-body > ul,

```

```

.timeline .timeline-body > p {
  margin-bottom: 0;
}

@media (min-width: 768px) {
  .timeline:before {
    left: 50%;
  }
  .timeline > li {
    min-height: 100px;
    margin-bottom: 100px;
  }
  .timeline > li .timeline-panel {
    float: left;
    width: 41%;
    padding: 0 20px 20px 30px;
    text-align: right;
  }
  .timeline > li .timeline-image {
    left: 50%;
    width: 100px;
    height: 100px;
    margin-left: -50px;
  }
  .timeline > li .timeline-image h4 {
    font-size: 13px;
    line-height: 18px;
    margin-top: 16px;
  }
  .timeline > li.timeline-inverted > .timeline-panel {
    float: right;
    padding: 0 30px 20px 20px;
    text-align: left;
  }
}

@media (min-width: 992px) {
  .timeline > li {
    min-height: 150px;
  }
  .timeline > li .timeline-panel {
    padding: 0 20px 20px;
  }
  .timeline > li .timeline-image {
    width: 150px;
    height: 150px;
    margin-left: -75px;
  }
  .timeline > li .timeline-image h4 {
    font-size: 18px;
    line-height: 26px;
    margin-top: 30px;
  }
  .timeline > li.timeline-inverted > .timeline-panel {
    padding: 0 20px 20px;
  }
}

@media (min-width: 1200px) {
  .timeline > li {
    min-height: 170px;
  }
  .timeline > li .timeline-panel {
    padding: 0 20px 20px 100px;
  }
  .timeline > li .timeline-image {
    width: 170px;
    height: 170px;
    margin-left: -85px;
  }
  .timeline > li .timeline-image h4 {
    margin-top: 40px;
  }
  .timeline > li.timeline-inverted > .timeline-panel {
    padding: 0 100px 20px 20px;
  }
}

#bidform{
  width: 60%;
  right: 0px;
  min-height: 560px;
  margin: 0px auto;
  width: 500px;
  position: relative;
}

a.hiddenanchor{

```

```

        display: none;
    }
    .text-muted1, .text-muted2 , .text-muted3{

        padding right: 5px;
        padding left:5px;
    }

```

JS

```

var timerduration= new Date().getTime()+60000; //choose anytime

//updating time function
var bidtimer = setInterval(function()
{
    // Working out timer from real time
    var currenttime = new Date().getTime();
    var time = timerduration - currenttime;

    //CALCULATIONS OF TIME

    var h = Math.floor((time % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    var m = Math.floor((time % (1000 * 60 * 60)) / (1000 * 60));
    var s = Math.floor((time % (1000 * 60)) / 1000);

    document.getElementById("timer1").innerHTML = h + "h "
    + m + "m " + s + "s ";

    if (time< 0)
    {
        clearInterval(bidtimer);
        document.getElementById("timer1").innerHTML = "Auction Finished";
        winningbid();
    }
}, 1000);

function bid() {

    var username = document.getElementById("username1").value;

    var bidvalue = document.getElementById("bidvalue1").value;
    console.log("bidvalue: " + bidvalue);
    //converting number
    var convertbidvalue = parseFloat(bidvalue);
    console.log("Convertbidvalue: " + convertbidvalue);

    // working out highest bid

    var minbid = document.getElementById("minbid1").value;
    var maxbid = document.getElementById("highbid1").value;
    console.log("maxbid: " + maxbid);

    //Displaying highest bid

    if (bidvalue > minbid && bidvalue > maxbid) {

        document.getElementById("highbid1").innerHTML = "Highest bid:" + bidvalue;
        document.getElementById("highbid1").value = bidvalue;
        postbid();

    } else {
        alert("Please enter a number higher than " + maxbid);
    }
}

function winningbid(){
    var winningbid = document.getElementById("highbid1").value;
    var email = document.getElementById("username1").value;

    if (winningbid == 0 || email == "")
    {
        alert("No winner!!")
    }
    else {
        alert("The winning bid was: " + winningbid + " by: " + email + " we'll be in touch shortly");
    }
}

function postbid() {

    var biddata = JSON.stringify({

        emailaddress: document.getElementById("username1").value,
        bidamount: document.getElementById("highbid1").value
    })

```

```

$.ajax({
    url: "/bidding",
    type: "POST",
    dataType: "json",
    contentType: "application/json",
    data: biddata,
    success: successbid,
    error: errorbid,
})

}

function successbid(){
    alert("Bid successful");
}

function errorbid(){
    alert("Error = bid not submitted");
}

$(document).ready(function () {
    document.getElementById("highbid1").value = 0;
    $("#submit1").click(function () {
        bid();
    });
    setInterval;
});

```

```

var timerduration= new Date().getTime()+60000; //choose anytime

//updating time function
var bidtimer = setInterval(function()
{
    // Working out timer from real time
    var currenttime = new Date().getTime();
    var time = timerduration - currenttime;

    //CALCULATIONS OF TIME

    var h = Math.floor((time % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    var m = Math.floor((time % (1000 * 60 * 60)) / (1000 * 60));
    var s = Math.floor((time % (1000 * 60)) / 1000);

    document.getElementById("timer2").innerHTML = h + "h "
    + m + "m " + s + "s ";

    if (time< 0)
    {
        clearInterval(bidtimer);
        document.getElementById("timer2").innerHTML = "Auction Finished";
        winningbid();
    }
}, 1000);

function bid() {

    var username = document.getElementById("username2").value;

    var bidvalue = document.getElementById("bidvalue2").value;
    console.log("bidvalue: " + bidvalue);
    //converting number
    var convertbidvalue = parseFloat(bidvalue);
    console.log("Convertbidvalue: " + convertbidvalue);

    // working out highest bid

    var minbid = document.getElementById("minbid2").value;
    var maxbid = document.getElementById("highbid2").value;
    console.log("maxbid: " + maxbid);

    //Displaying highest bid

    if (bidvalue > minbid && bidvalue > maxbid) {

        document.getElementById("highbid2").innerHTML = "Highest bid:" + bidvalue;
        document.getElementById("highbid2").value = bidvalue;
        postbid();

    } else {
        alert("Please enter a number higher than " + maxbid);
    }
}

function winningbid(){
    var winningbid = document.getElementById("highbid2").value;
    var email = document.getElementById("username2").value;

    if (winningbid == 0 || email == "")

```

```

        {
            alert("No winner!!")
        }
    else {
        alert("The winning bid was: " + winningbid + " by: " + email + " we'll be in touch shortly");
    }
}

function postbid() {

    var biddata = JSON.stringify({
        emailaddress: document.getElementById("username2").value,
        bidamount: document.getElementById("highbid2").value
    })
    $.ajax({
        url: "/bidding",
        type: "POST",
        dataType: "json",
        contentType: "application/json",
        data: biddata,
        success: successbid,
        error: errorbid,
    })
}

function successbid(){
    alert("Bid successful");
}

function errorbid(){
    alert("Error = bid not submitted");
}

$(document).ready(function () {
    document.getElementById("highbid2").value = 0;
    $("#submit2").click(function () {
        bid();
    });
    setInterval;
});

```

```

var timerduration= new Date().getTime()+60000; //choose anytime

//updating time function
var bidtimer = setInterval(function()
{
    // Working out timer from real time
    var currenttime = new Date().getTime();
    var time = timerduration - currenttime;

    //CALCULATIONS OF TIME

    var h = Math.floor((time % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    var m = Math.floor((time % (1000 * 60 * 60)) / (1000 * 60));
    var s = Math.floor((time % (1000 * 60)) / 1000);

    document.getElementById("timer3").innerHTML = h + "h "
    + m + "m " + s + "s ";

    if (time< 0)
    {
        clearInterval(bidtimer);
        document.getElementById("timer3").innerHTML = "Auction Finished";
        winningbid();
    }
}, 1000);

function bid() {

    var username = document.getElementById("username3").value;

    var bidvalue = document.getElementById("bidvalue3").value;
    console.log("bidvalue: " + bidvalue);
    //converting number
    var convertbidvalue = parseFloat(bidvalue);
    console.log("Convertbidvalue: " + convertbidvalue);

    // working out highest bid

    var minbid = document.getElementById("minbid3").value;
    var maxbid = document.getElementById("highbid3").value;
    console.log("maxbid: " + maxbid);

    //Displaying highest bid

```

```

    if (bidvalue > minbid && bidvalue > maxbid) {

        document.getElementById("highbid3").innerHTML = "Highest bid:" + bidvalue;
        document.getElementById("highbid3").value = bidvalue;
        postbid();

    } else {
        alert("Please enter a number higher than " + maxbid);
    }
}
function winningbid(){
    var winningbid = document.getElementById("highbid3").value;
    var email = document.getElementById("username3").value;

    if (winningbid == 0 || email == "")
    {
        alert("No winner!!")
    }
    else {
        alert("The winning bid was: " + winningbid + " by: " + email + " we'll be in touch shortly");
    }
}

function postbid() {

    var biddata = JSON.stringify({

        emailaddress: document.getElementById("username3").value,
        bidamount: document.getElementById("highbid3").value
    })
    $.ajax({
        url: "/bidding",
        type: "POST",
        dataType: "json",
        contentType: "application/json",
        data: biddata,
        success: successbid,
        error: errorbid,
    })

}
function successbid(){
    alert("Bid successful");
}
function errorbid(){
    alert("Error = bid not submitted");
}

$(document).ready(function () {
    document.getElementById("highbid3").value = 0;
    $("#submit3").click(function () {
        bid();
    });
    setInterval;
});

```

```

function getminbid(batchid){
    $.ajax({
        url: "/batch/" + batchid,
        type: "GET",
        dataType: "json",

    }).then(function(data) {
        var minbid = data.minbatchprice;
        console.log(minbid);
        document.getElementById("minbid1").value = minbid;
        document.getElementById("minbid1").innerHTML = "Minimum bid: " +minbid;
    })
}

$(document).ready(function () {

    getminbid(1);
});

```

```

function getminbid(batchid){
    $.ajax({
        url: "/batch/" + batchid,
        type: "GET",
        dataType: "json",

    }).then(function(data) {
        var minbid = data.minbatchprice;
        console.log(minbid);
    })
}

```

```

        document.getElementById("minbid2").value = minbid;
        document.getElementById("minbid2").innerHTML = "Minimum bid: " + minbid;
    })
}

$(document).ready(function () {

    getminbid(2);
});

```

```

function getminbid(batchid){
    $.ajax({
        url: "/batch/" + batchid,
        type: "GET",
        dataType: "json",

        }).then(function(data) {
            var minbid = data.minbatchprice;
            console.log(minbid);
            document.getElementById("minbid3").value = minbid;
            document.getElementById("minbid3").innerHTML = "Minimum bid: " + minbid;
        })
}

$(document).ready(function () {

    getminbid(3);
});

```

```

function showbatch(batchid){
    console.log(batchid);
    $.ajax({
        url: "/batch/" + batchid + "/regList",
        type: "GET",
        dataType: "json",
        success: function(res) {
            if (batchid == 1){$(".text-muted1").html(""); //Clear contents of adding div tag
            var htmlContent = "";
            htmlContent = "<br>";
            res._embedded.reg.forEach(function (element) {
                console.log("regplate: " + element.regplate);
                htmlContent += element.regplate + " ";
                $(".text-muted1").html(htmlContent); // Rewrite contents of adding div tag
            })
            else if (batchid == 2){
            $(".text-muted2").html(""); //Clear contents of adding div tag
            var htmlContent = "";
            htmlContent = "<br>";
            res._embedded.reg.forEach(function (element) {
                console.log("regplate: " + element.regplate);
                htmlContent += element.regplate + " ";
                $(".text-muted2").html(htmlContent);
                // Rewrite contents of adding div tag
            })
            else if (batchid == 3){
            $(".text-muted3").html(""); //Clear contents of adding div tag
            var htmlContent = "";
            htmlContent = "<br>";
            res._embedded.reg.forEach(function (element) {
                console.log("regplate: " + element.regplate);
                htmlContent += element.regplate + " ";
                $(".text-muted3").html(htmlContent);
                // Rewrite contents of adding div tag
            })
        }
    });
}

$(document).ready(function () {

    $("#batch1").click(function(){showbatch(1);});
    $("#batch2").click(function(){showbatch(2);});
    $("#batch3").click(function(){showbatch(3);});

});

```

pom

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.3.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>webassignment</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>webassignment</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-rest</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.postgresql</groupId>
      <artifactId>postgresql</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```