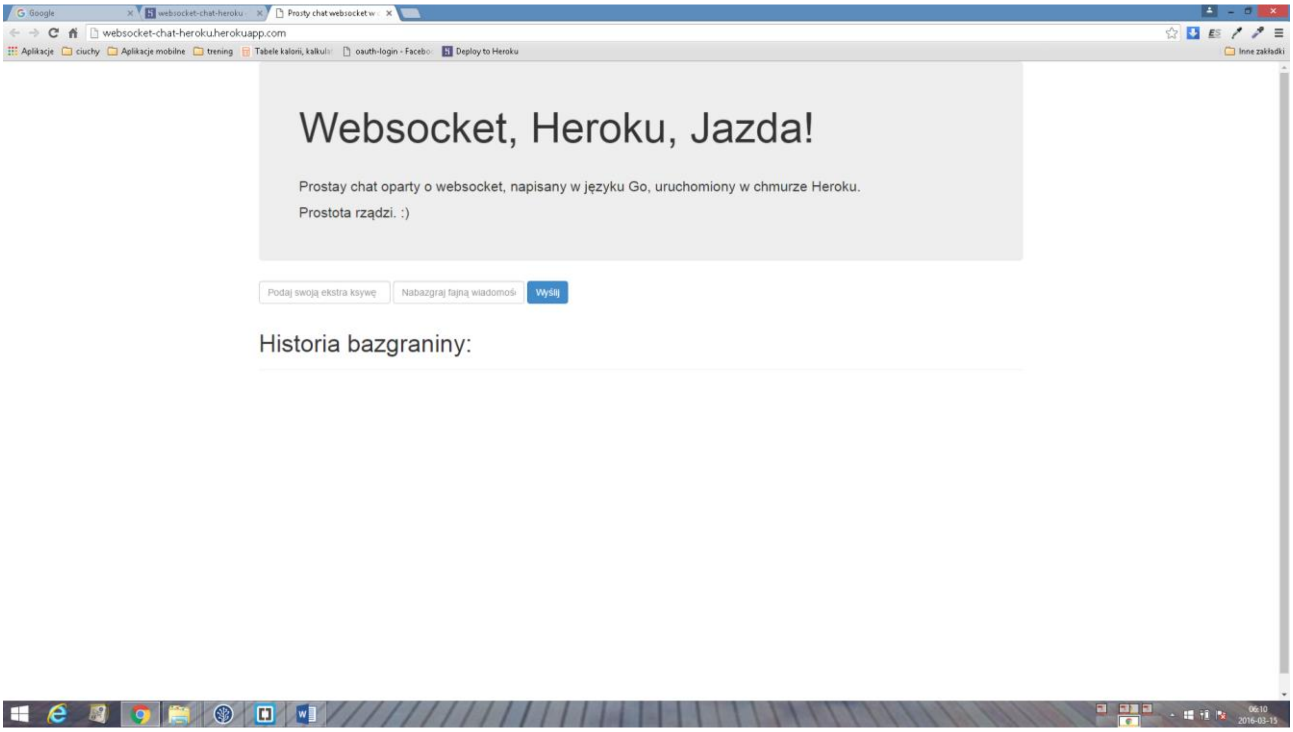
OOR: Programowanie równoległe i rozproszone

Laboratorium 7:

Cloud Computing



Websocket, Heroku, Jazda!

Prostay chat oparty o websocket, napisany w języku Go, uruchomiony w chmurze Heroku.

Prostota rządzi. :)

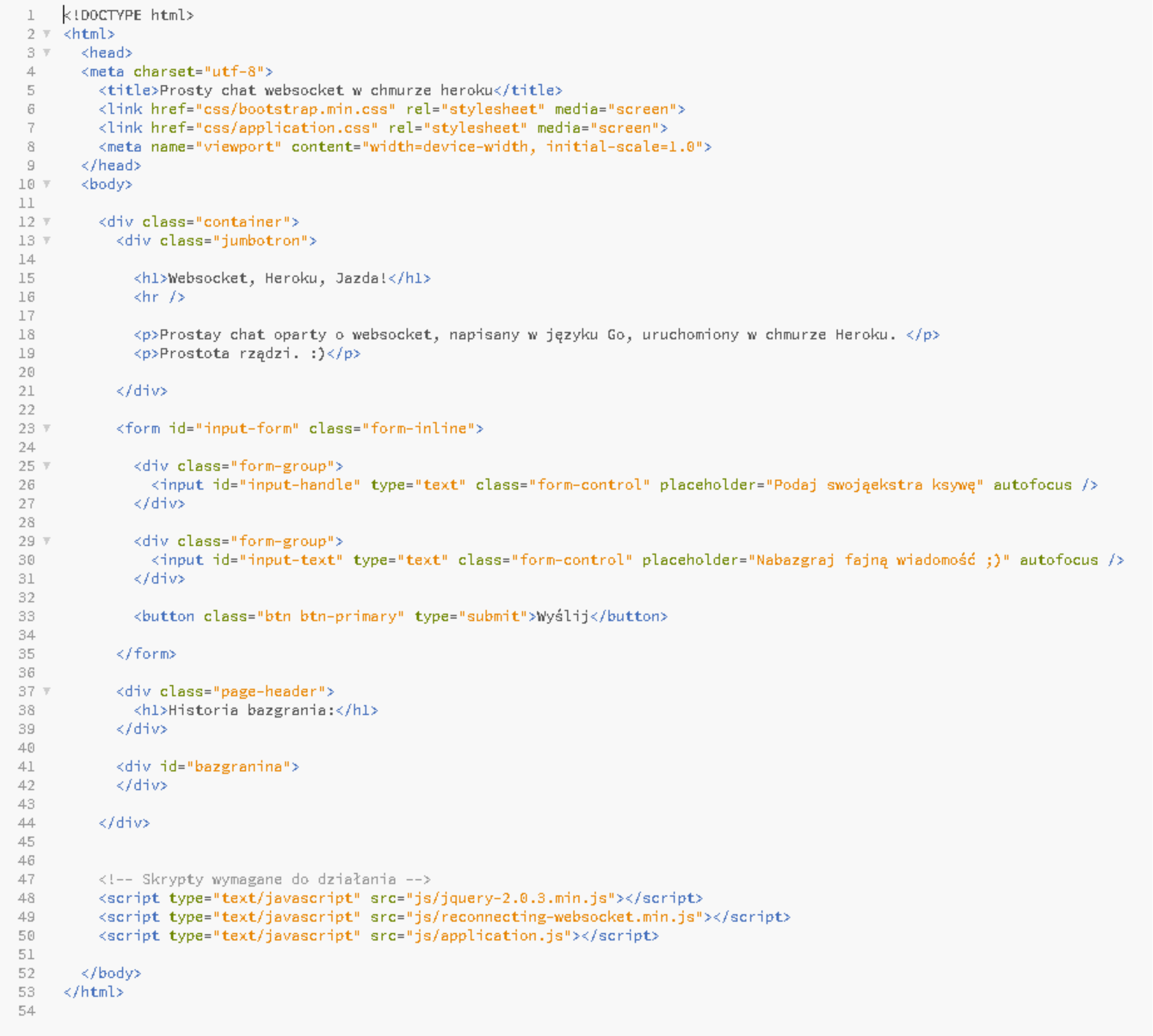
Historia bazgraniny:

Lab_6
125NCI – Stanisław Siemaszko

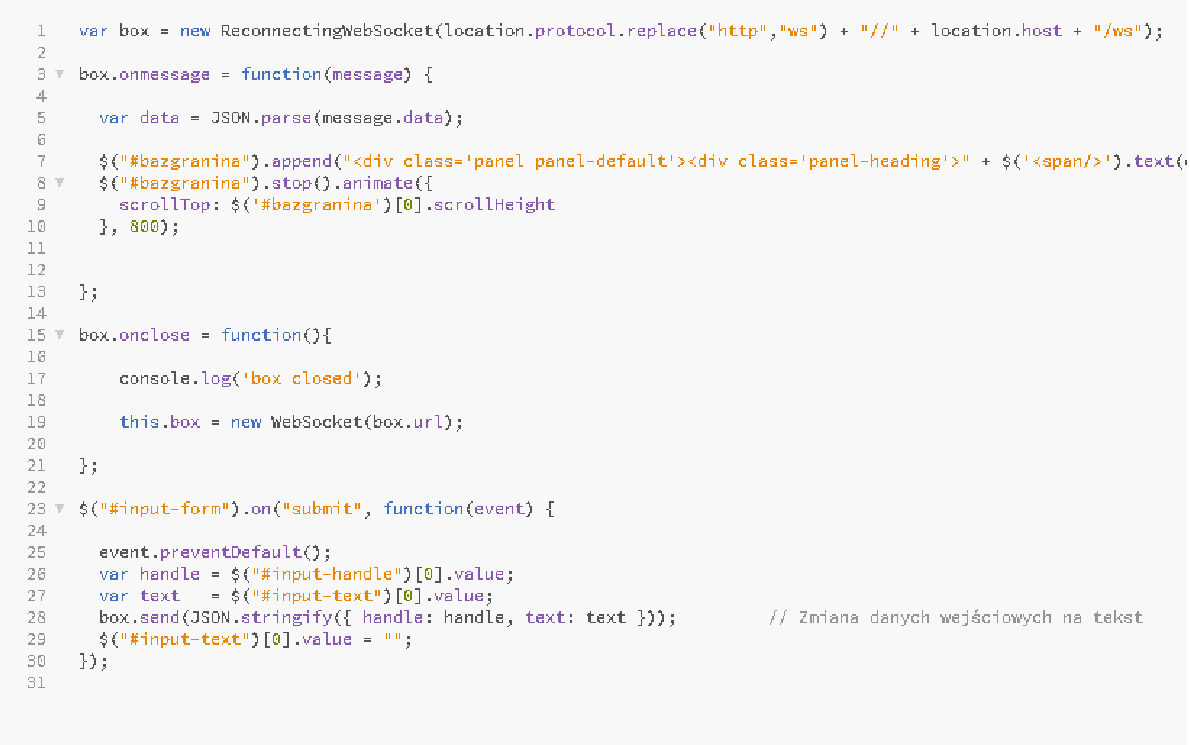
Jest to prosta aplikacja działająca w chmurze heroku, napisana w języku Go, obsługująca komunikację przy użyciu Websocket.

Aplikacja dostępna jest pod adresem: http://websocket-chat-heroku.herokuapp.com/

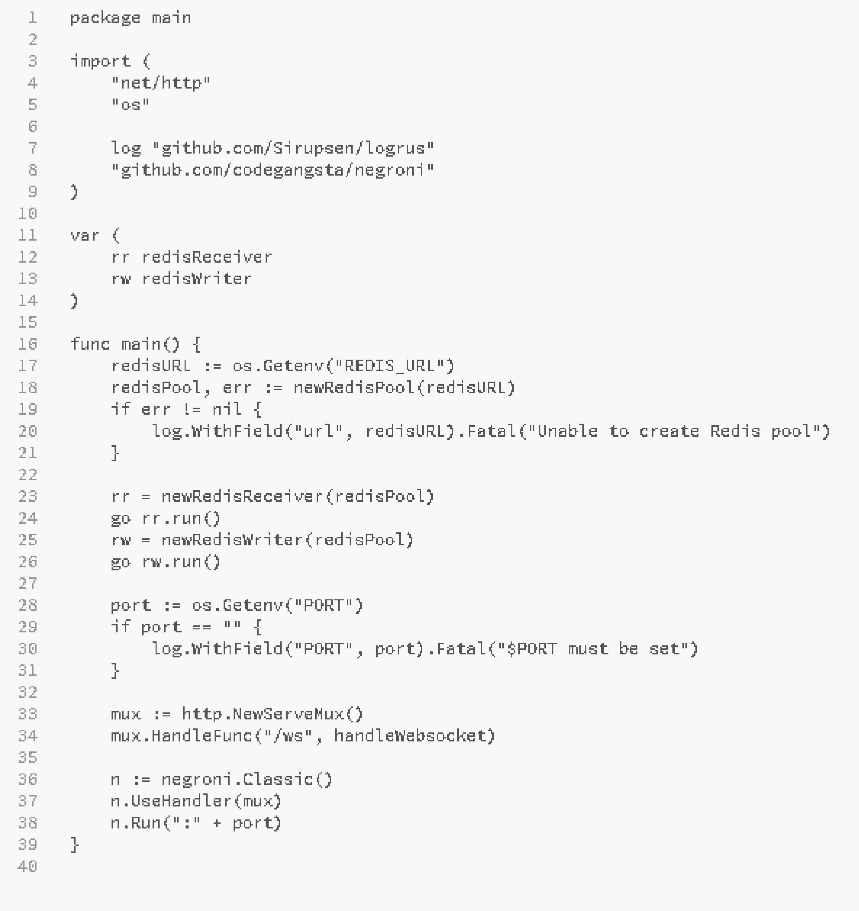
Kod aplikacji:



```html
<!DOCTYPE html>
<html>
  <head>
  <meta charset="utf-8">
    <title>Prosty chat websocket w chmurze heroku</title>
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
    <link href="css/application.css" rel="stylesheet" media="screen">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>

    <div class="container">
      <div class="jumbotron">

        <h1>Websocket, Heroku, Jazda!</h1>
        <hr />

        <p>Prostay chat oparty o websocket, napisany w języku Go, uruchomiony w chmurze Heroku. </p>
        <p>Prostota rządzi. :)</p>

      </div>

      <form id="input-form" class="form-inline">

        <div class="form-group">
          <input id="input-handle" type="text" class="form-control" placeholder="Podaj swojąekstra ksywę" autofocus />
        </div>

        <div class="form-group">
          <input id="input-text" type="text" class="form-control" placeholder="Nabazgraj fajną wiadomość ;)" autofocus />
        </div>

        <button class="btn btn-primary" type="submit">Wyślij</button>

      </form>

      <div class="page-header">
        <h1>Historia bazgrania:</h1>
      </div>

      <div id="bazgranina">
      </div>

    </div>


    <!-- Skrypty wymagane do działania -->
    <script type="text/javascript" src="js/jquery-2.0.3.min.js"></script>
    <script type="text/javascript" src="js/reconnecting-websocket.min.js"></script>
    <script type="text/javascript" src="js/application.js"></script>

  </body>
</html>

```



```javascript
var box = new ReconnectingWebSocket(location.protocol.replace("http","ws") + "//" + location.host + "/ws");

box.onmessage = function(message) {

  var data = JSON.parse(message.data);

  $("#bazgranina").append("<div class='panel panel-default'><div class='panel-heading'>" + $('<span/>').text(
  $("#bazgranina").stop().animate({
    scrollTop: $('#bazgranina')[0].scrollHeight
  }, 800);


};

box.onclose = function(){

    console.log('box closed');

    this.box = new WebSocket(box.url);

};

$("#input-form").on("submit", function(event) {

  event.preventDefault();
  var handle = $("#input-handle")[0].value;
  var text   = $("#input-text")[0].value;
  box.send(JSON.stringify({ handle: handle, text: text }));         // Zmiana danych wejściowych na tekst
  $("#input-text")[0].value = "";
});

```



```go
package main

import (
    "net/http"
    "os"

    log "github.com/Sirupsen/logrus"
    "github.com/codegangsta/negroni"
)

var (
    rr redisReceiver
    rw redisWriter
)

func main() {
    redisURL := os.Getenv("REDIS_URL")
    redisPool, err := newRedisPool(redisURL)
    if err != nil {
        log.WithField("url", redisURL).Fatal("Unable to create Redis pool")
    }

    rr = newRedisReceiver(redisPool)
    go rr.run()
    rw = newRedisWriter(redisPool)
    go rw.run()

    port := os.Getenv("PORT")
    if port == "" {
        log.WithField("PORT", port).Fatal("$PORT must be set")
    }

    mux := http.NewServeMux()
    mux.HandleFunc("/ws", handleWebsocket)

    n := negroni.Classic()
    n.UseHandler(mux)
    n.Run(":" + port)
}

```