

Corso di Tecnologie e Linguaggi per il Web

A.A. 2018-2019

Over Paste **(Paste your every idea)**

Andrea Draghetti

Autore:	Andrea Draghetti
Ultima modifica:	19 febbraio 2019 - versione 1.0
Prima modifica:	19 febbraio 2019

1) Introduzione

Il progetto denominato “Over Paste” vuole essere una alternativa al noto portale PasteBin dove è possibile inviare frammenti di testo, solitamente di codice sorgente, per la condivisione pubblica.

Il progetto è stato reso pubblico sulla nota piattaforma GitHub: <https://github.com/drego85/OverPaste>

Breve analisi dei requisiti

a) Destinatari

All’utente che utilizza l’applicazione non sono richieste particolari capacità o competenze tecniche, non è altresì richiesto l’uso di specifici device (computer, smartphone o tablet). Tuttavia si presume che i fruitori dell’applicazione, vista la funzionalità di condivisione di codice sorgente, siano utenti avanzati e che utilizzano un ambiente desktop per la condivisione dei propri progetti.

Infine la banda richiesta è veramente bassa in quanto non vi è una grande quantità di informazioni da trasmettere.

b) Linguaggio

L’applicazione non fa uso di particolare terminologia tecnica o settoriale perciò risulta fruibile dall’utente medio senza che siano necessarie particolari competenze linguistiche.

c) Modello di valore

L’obiettivo è quello di fornire un supporto ad uno sviluppatore o ad un team di sviluppatori che vuole condividere con semplicità porzioni di codice sorgente da revisionare o migliorare. È un valore aggiunto poiché permette una rapida condivisione del proprio codice che può essere divulgato grazie ad un link via eMail, Social Network o Instant Message.

d) Flusso dei dati

Per mantenere la persistenza dei dati e rendergli visionabili con diversi dispositivi i paste vengono salvati su un database non relazionale MongoDB.

e) Ottenere i contenuti.

I contenuti vengono salvati in un database non relazionale MongoDB; tale database sfrutta le specifiche del movimento NoSQL che promuove software dove la persistenza dei dati è caratterizzata dal fatto di non utilizzare il modello relazionale.

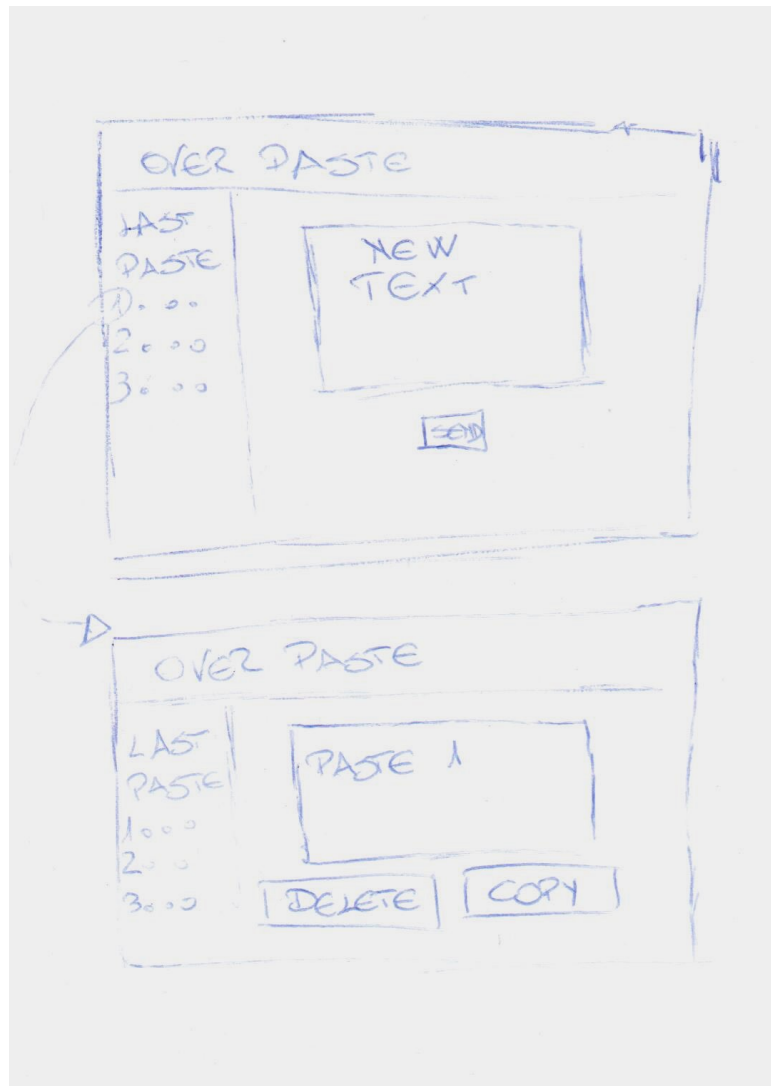
f) Aspetti tecnologici

Per la realizzazione di Over Paste sono state sfruttate diverse tecnologie che andrò a riepilogare di seguito:

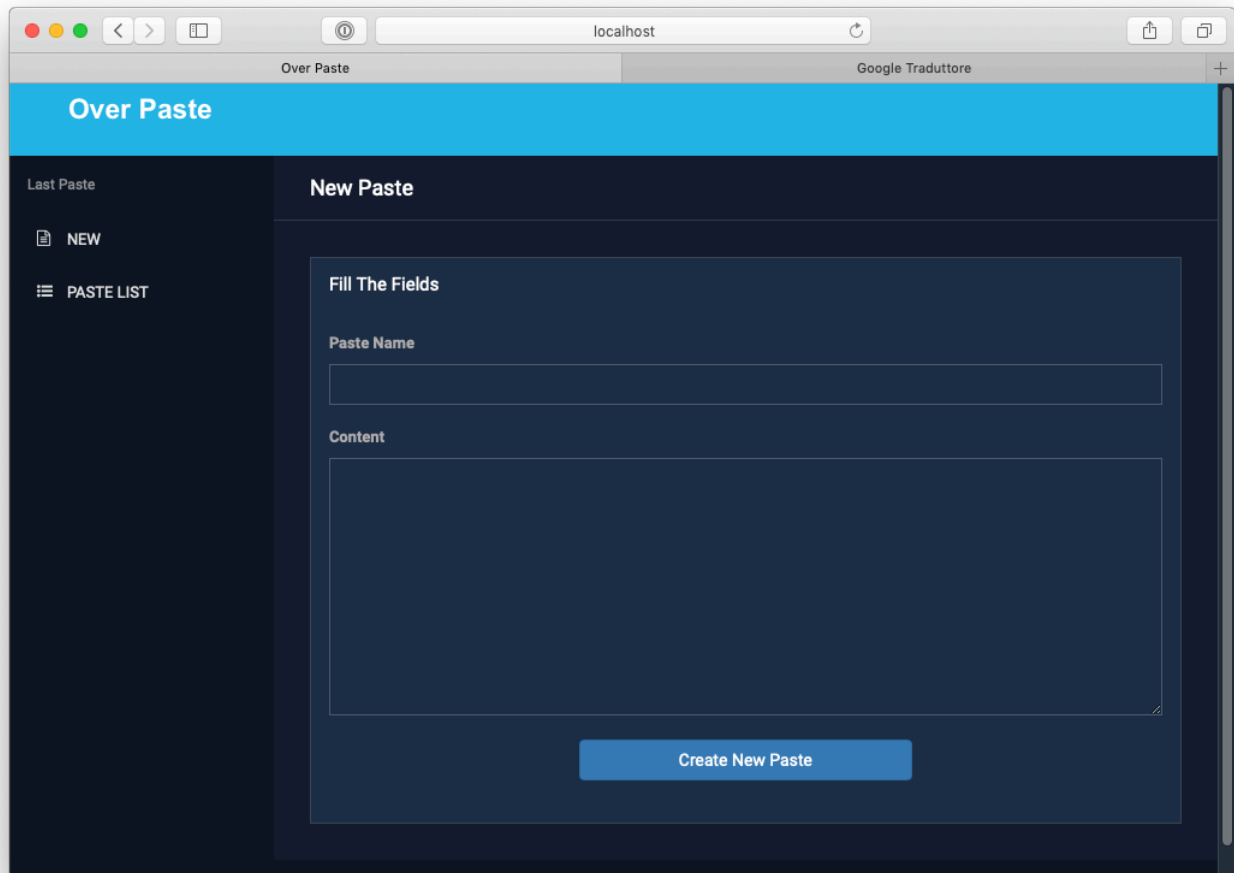
- NodeJS ovvero un runtime JavaScript costruito sul motore di JavaScript V8 di Chrome, sfruttato come backend;
- Express un framework web veloce, non categorico e minimalista per NodeJS;
- Handlebarsjs un template engine per NodeJS;
- Bootstrap un toolkit opensource per sviluppare in HTML, CSS e JS;
- JQuery una libreria Javascript;
- Mongoose un plugin per il collegamento a MongoDB basato su schemi e modelli;
- API HTML5 sfruttando la nuova funzionalità di clipboard introdotta in HTML5 per copiare i paste.

2) Interfacce

1. Prototipo



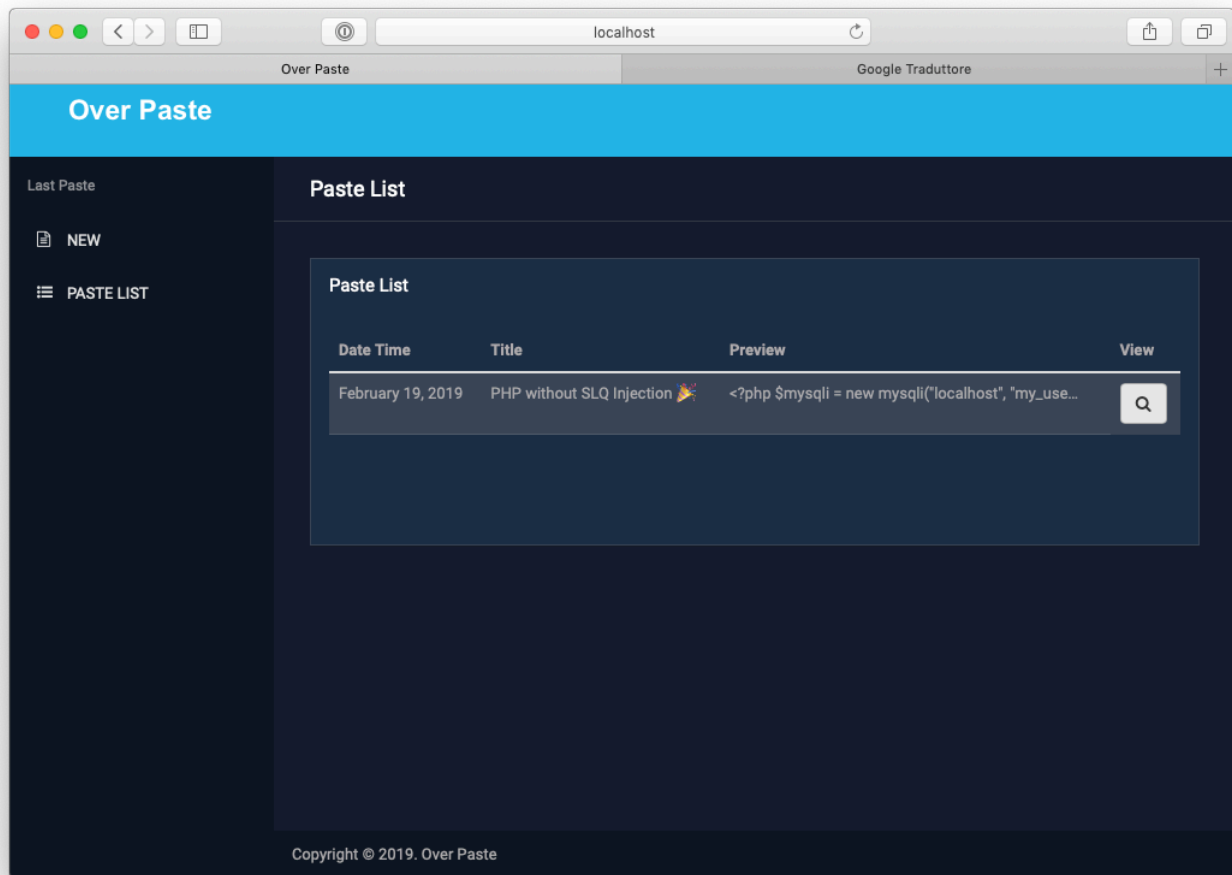
2. Web browser



Homepage

Direttamente dalla pagina principale è possibile procedere alla creazione di un nuovo paste indicando il titolo di esso e il relativo contenuto, inoltre tramite il menu laterale è possibile accedere alla lista dei paste già esistenti.

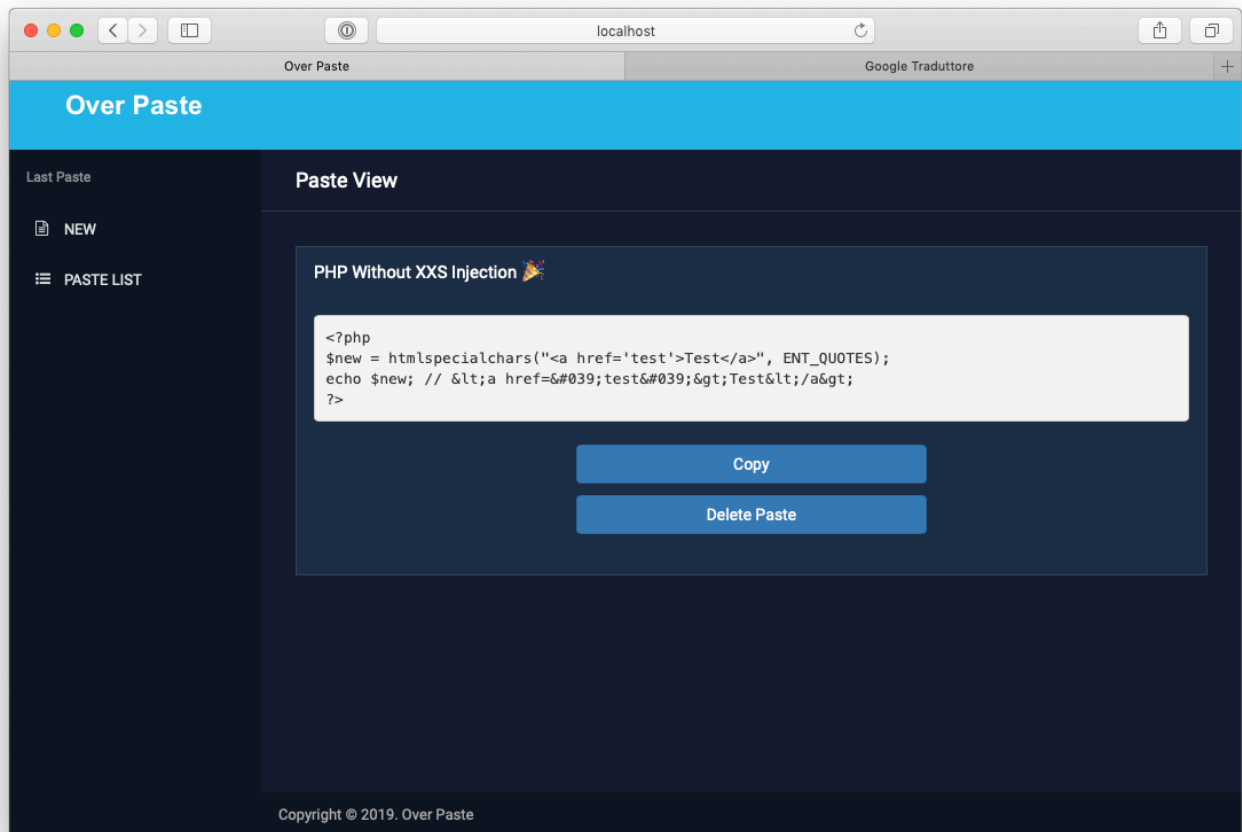
List



La Paste List permette di visualizzare in ordine cronologico i paste creati mostrando per ogni singolo paste:

- Data di creazione;
- Titolo;
- Preview, grazie alla funzione di Ellipsis inclusa nel modulo handlebars-helpers vengono estratti i primi 50 caratteri del paste.

Infine selezionando sulla lente di ingrandimento per ogni singolo paste è possibile accedere alla visualizzazione di esso.



Visualizzazione Paste

Nella visualizzazione di un singolo paste viene mostrato il titolo e successivamente il contenuto di esso all'interno di un campo HTML `<pre>`.

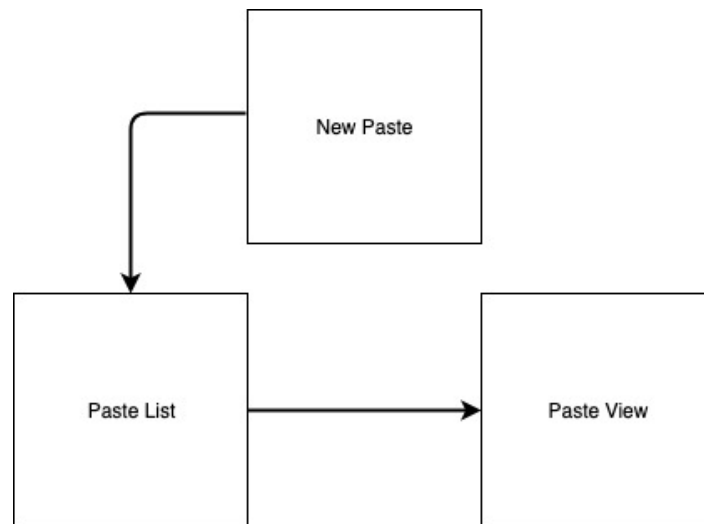
È inoltre possibile copiare il contenuto del paste semplicemente premendo il pulsante "Copy", il quale sfrutterà le API di HTML5 per copiare in clipboard il testo.

Infine attraverso il pulsante "Delete Paste" è possibile eliminare il paste.

3) Architettura

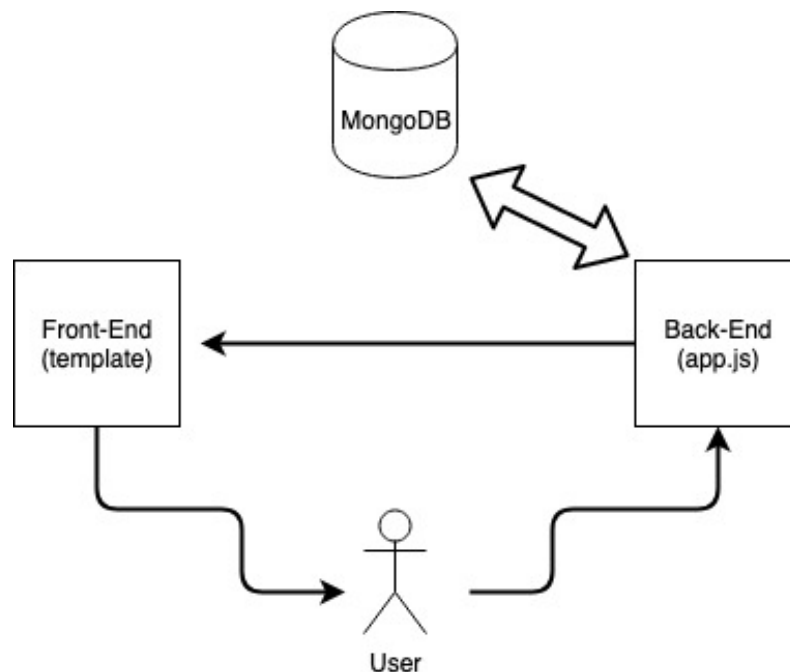
1. Diagramma dell'ordine gerarchico delle risorse

Di seguito si evidenzia il diagramma ordinato delle risorse disponibili nell'applicazione.



2. Descrizione delle risorse

Di seguito è sinteticamente visibile il diagramma delle risorse sfruttate per la realizzazione dell'applicazione, il Back-End in NodeJS ha il compito di dialogare con il database MongoDB e di rappresentare mediante il Front-End i dati archiviati, ed inoltre di ricevere mediante richieste HTTP GET o POST i dati trasmessi dall'utente.



3. Struttura Database

Il database è composto dai seguenti record

Colonna	Tipo	Descrizione
id	Int	Identificativo univoco del record
title	String	Titolo del Paste
pastetext	String	Contenuto del Paste
datetime	Date	Orario di inserimento del paste

4) Codice

Frammenti del codice più significativo

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Over Paste</title>
5     <meta charset="utf-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <meta name="description" content="Paste your every idea!">
9     <meta name="keywords" content="Paste, PasteBin">
10    <meta name="author" content="Andrea Draghetti">
11
12    <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">
13    <link rel="icon" href="images/favicon.ico" type="image/x-icon">
14
15    <!-- Common plugins -->
16    <link href="plugins/bootstrap/css/bootstrap.min.css" rel="stylesheet">
17    <link href="plugins/font-awesome/css/font-awesome.min.css" rel="stylesheet">
18    <link rel="stylesheet" href="plugins/metisMenu/metisMenu.min.css">
19
20    <!-- Template CSS -->
21    <link href="css/style.css" rel="stylesheet">
22
23  </head>
24  <body>
25
26    <!--top bar start-->
27    <div class="top-bar bg-primary">
28      <div class="container-fluid">
29        <div class="row">
30          <div class="col-xs-6">
31            <a href="/" class="top-logo">
32              <h1></h1>
33            </a>
34            <div class="left-nav-toggle visible-xs visible-sm">
35              <a href="">
36                <i class="glyphicon glyphicon-menu-hamburger"></i>
37              </a>
38            </div><!--end nav toggle icon-->
39          </div>
40        </div>
41      </div>
42    </div>
43    <!-- top bar end-->
44
45    <!--left navigation start-->
46    <aside class="float-navination">

```

Porzione del Main del Front-End che viene gestito dal Template Engine Handlebarsjs.

```

1  const mongoose = require("mongoose");
2
3  const pasteSchema = mongoose.Schema({
4    title:{
5      type: String,
6      required: true
7    },
8    pastetext:{
9      type: String,
10     required: true
11   },
12   datetime:{
13     type: Date,
14     default: Date.now
15   }
16 });
17
18 mongoose.model("Notes" , pasteSchema);

```

Schema e Modello del database

```

2 function copy(){
3     var text = document.getElementById("pastetext");
4     var range = document.createRange();
5
6     range.selectNode(text);
7     window.getSelection().addRange(range);
8     document.execCommand("copy");
9
10 }

```

Funzione Javascript per la copia del contenuto del Paste in clipboard

```

<tr>
  <td>{{moment data format="YYYY-MM-DD HH:mm"}}</td>
  <td>{{title}}</td>
  <td>{{ellipsis pastetext 50}}</td>
  <td class="center"><a href="/view_paste/{{id}}"><button type="button" class="btn
</tr>

```

Utilizzo di Ellipsis e Moment per la formattazione del testo e della data.

```

70 // Route for Paste Views
71 app.get("/view_paste/:id", function(req, res, next){
72   console.log("Request URL:", req.originalUrl);
73   let headerTitle = "Paste View";
74   Notes.findOne({
75     _id: req.params.id
76   })
77   .then(paste =>{
78     res.render("view_paste" , {
79       paste: paste,
80       headerTitle: headerTitle
81     });
82   });
83 });
84
85 // Route to save New Paste
86 app.post("/", function(req, res, next){
87   const newPaste = {
88     title: req.body.title,
89     pastetext: req.body.pastetext
90   }
91   new Notes(newPaste)
92   .save()
93   .then(nota =>{
94     res.redirect("/list_paste");
95   })
96 });
97

```

Routing per la visualizzazione e il salvataggio di un nuovo paste

5) Conclusioni

In conclusione posso dire che il progetto è stato interessante, poiché mi ha dato la possibilità di conoscere nuovi linguaggi (NodeJS), framework (Express) e template engine (Handlebarsjs) che personalmente non avevo mai sfruttato. Ho trovato di notevole l'interesse il funzionamento di NodeJS che grazie alla realizzazione di un linguaggio di Input-Output *non bloccante* risulta essere più competitivo rispetto a linguaggi storici come PHP.

6) Nota bibliografica e sitografica

- [NodeJS](#)
- [MongoDB](#)
- [Express](#)
- [Handlebarsjs](#)
- [Bootstrap](#)