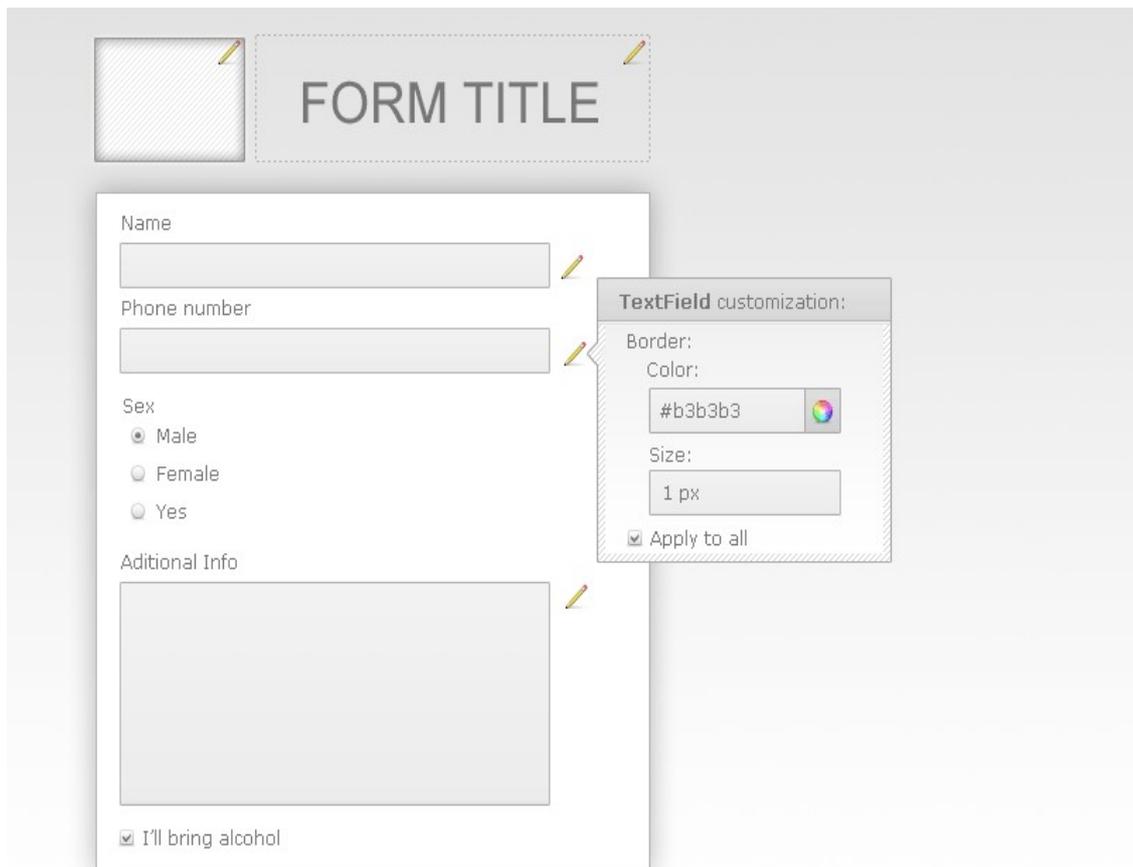


# **Theme Editor**

Emanuele Ricci, Ivan Litsvinenka, Simone Della Longa

# ThemeEditor Logic



The ThemeEditor allow the user to customize each fields in his personal form layout. Each field is identified by a CSS's class and id.

On the right side of the input field there's an icon that allow the user to customize some properties of the field ( the set of properties is defined by Fields Group ). The user can choose if apply that change at the selected input field ( using the ID ) or to each fields that belong to that CSS class.

Also than fields customization the user can upload an image to associate to the form and can personalize it with a title.

With ThemeEditor the user can save his theme and re-use it with others forms that he will create.

# Property List Structure

```
var textPropertyList = [
  {
    "group": "border",
    "properties": [
      {
        "name": "border-width",
        "type": "text",
        "validator": "size",
        "info": "Width of Field Border. Example: 2px"
      },
      {
        "name": "border-color",
        "type": "text",
        "validator": "color",
        "info": "Color of Field Border. Example: #ff00ff"
      },
      {
        "name": "border-style",
        "type": "text",
        "validator": "none",
        "info": "Type of Field Border. Example: solid"
      }
    ]
  },
  {
    "group": "background",
    "properties": [
      {
        "name": "background-color",
        "type": "text",
        "validator": "color",
        "info": "Color of Field background. Example: #ff00ff"
      }
    ]
  }
]
```

This is an example of the properties binded on a field following the JSON structure. Every properties are grouped by a group name ( like 'border' ). and are composed by 4 fields:

- name: the name to be displayed on the page
- type: the input type for that properties
- validator: the type of validation
- info: the information to show to the user for help purpose