

## Relazione progetto Tecnologie Web (2017)

Il progetto è recuperabile all'url: <https://gitlab2.educ.di.unito.it/st149043/Tweb/tree/master/progetto>

Il nome del sito è “Rurals Community”, l'idea è quella di fornire ad una comunità rurale una semplice piattaforma tramite la quale i pochi abitanti della comunità possano coordinarsi e organizzarsi riguardo alle necessità dei compaesani, sia per quanto riguarda materie prime che per servizi.

### Home:

Lo schema del sito è molto lineare. Un utente che arriva sulla home del sito può visualizzare, oltre alla barra di navigazione, delle immagini (slideshow automatico), un articolo e una lista di eventi della comunità. Tentare di accedere alle altre sezioni risulterà in un redirect alla pagina di login.

### Login:

La pagina di login è divisa in due sezioni, una per effettuare l'accesso e una per registrarsi al sito. Il form da compilare è semplice (nome, cognome, username, password, conferma password e email opzionale). Un utente che effettua la registrazione potrà subito effettuare il login. Effettuare il login risulta in un redirect alla pagina del profilo personale. Una volta effettuato il login al posto della pagina di login l'utente visualizzerà la scritta “Logout”.

### Profile:

L'utente può vedere le proprie informazioni personali e una lista di eventi da lui creati. In caso non ci fossero eventi disponibili il sito invita l'utente a crearne uno.

### Events:

In questa pagina l'utente può creare un evento, specificando nome dell'evento, data, e una breve descrizione. Nel caso invece avesse intenzione di eliminare un evento potrà farlo nella sezione sottostante.

### Weather:

L'utente inoltre può consultare delle previsioni meteo per la settimana corrente\*. Selezionando il giorno a cui è interessato verranno visualizzate le temperature e la probabilità di precipitazione.

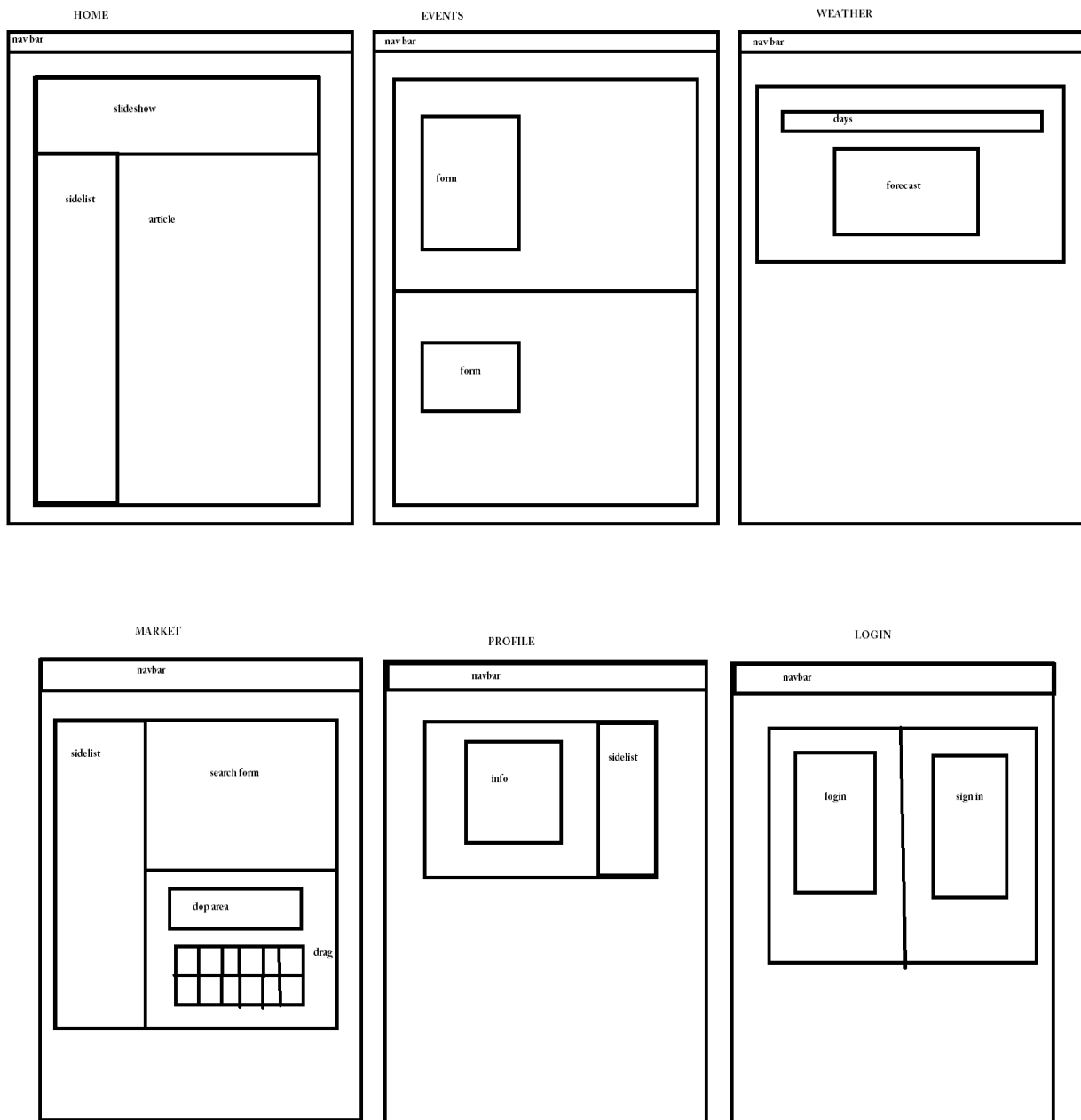
### Market:

L'ultima e più interessante parte del sito è il marketplace, ovvero la pagina tramite cui gli abitanti della comunità rurale possono organizzarsi e aiutarsi a vicenda. La pagina presenta una lista di richieste della gente, in caso non ce ne fossero il sito invita l'utente a richiedere qualcosa ai suoi compaesani. La sezione principale della pagina è un form utilizzabile dall'utente per specificare parametri per ricercare richieste presenti nel database (sia di altri che le sue, e anche già risolte). Se l'utente clicca su uno dei risultati potrà comunicare di essere disposto ad aiutare con quella specifica richiesta.

Una volta inviata la ricerca la pagina viene aggiornata mostrando i risultati della ricerca.

La sezione inferiore della pagina è utilizzata per effettuare richieste. Sono presenti i prodotti e servizi erogabili al momento e trascinando l'immagine del prodotto/servizio in un'area apposita e specificando la quantità l'utente può aggiungere una nuova richiesta al database.

## Mock-up:



## Parte tecnica:

### Front-end:

Vengono usate classi specifiche definite nei fogli di stile (css) per le regole stilistiche più usate (esempio padding: 1%) l'obiettivo è quello di usare il meno possibile regole per singoli elementi. Il contenuto delle pagine index.php e weather.php è recuperato tramite richieste Ajax (file xml), le sidelist sono elementi recuperati dal database, come in profile.php e come lo sono i risultati della

ricerca effettuabile da market.php. Il resto è contenuto html.

Un insieme di funzioni javascript gestisce gli eventi e il comportamento clientside.

- highlight della sezione corrente
- conta chars rimanenti in creazione evento
- rivelazione div nascosto “ricerca per nome” se “everyone” è checked
- drag and drop di elementi per effettuare richieste
- aggiornamento sezione meteo a seconda del giorno selezionato
- slideshow
- check formato data tramite regExp

I form vengono controllati client-side dagli appositi attributi forniti da html5. Link ed elementi con cui l'utente può interagire reagiscono all'azione hover tramite opportune regole css. Il codice javascript fa uso delle librerie prototype e scripacious, tutti i file js sono organizzati nella cartella /utility/js

### **Back-end:**

Il sito utilizza funzioni php specificate e documentate nei files utility/functions/utls.php e utility/functions/db.php, questultimo fornisce un'ampia collezione di funzioni per interrogare e modificare il database. Il database associativo è composto da:

- users (id\*, fname, lname, username, passwd, email)
- products(id\*, name, type, description, img)
- requests(id\*, id\_product, id\_buyer, id\_seller, quantity)
- events(id\*, id\_manager, event\_date, title, description)

Sono presenti due file .sql nella cartella “AA\_files database”, uno per la creazione del database con lo stretto necessario e uno per popolarlo di entry casuali per rendere più verosimile il testing.

Tutte le azioni dei form sono gestite con php, il nome file per le azioni ha formato “a\_nomeazione.php” e tutti i files di questo tipo sono organizzati nella cartella utility/actions e sono:

-a\_eventcreate: Chiamata dalla pagina events.php durante la creazione di un evento.

Richiede: un titolo, una data (formato yyyy-mm-dd) e una descrizione.

Metodo: POST

Successo: Inserisce un nuovo evento nel database

-a\_eventdelete: Chiamata dalla pagina events.php durante la cancellazione di un evento.

Richiede: un id evento

Metodo: POST

Successo: elimina l'evento specificato se chi lo ha richiesto è autorizzato

-a\_login: Chiamata dalla pagina login.php durante il login.

Richiede: username e password

Metodo: POST

Successo: viene creata una sessione valida

-a\_logout: Chiamata dal link nella barra di navigazione.

Richiede: /

Metodo: /

Successo: distrugge la sessione e ne crea una nuova

-a\_requestcreate: Chiamata dalla pagina market.php quando si effettua un nuovo ordine.

Richiede: Quantità e nome prodotto

Metodo: POST

Successo: inserisce una nuova richiesta nel database

- a\_requestsatisfy: Chiamata dalla pagina market.php quando viene soddisfatta una richiesta
  - Richiede: id della richiesta
  - Metodo: GET
  - Successo: aggiorna id\_seller della richiesta corrispondente
- a\_signin: Chiamata dalla pagina login.php quando viene creato un nuovo utente.
  - Richiede: i campi obbligatori del form di signin
  - Metodo: POST
  - Successo: Viene aggiunto un nuovo utente al sito

Tutte le azioni comunicano con un messaggio nella pagina html (non intrusivo) il risultato dell'operazione, sia in caso positivo che negativo (normalmente intercettato prima da js o dagli attributi html5).

Comunicazione f/b-end:

La maggior parte delle comunicazioni avviene tramite php. Una funzione recupera le informazioni e vengono poi inserite nella pagina. Le uniche due richieste Ajax sono nella pagina index.php e weather.php. Le chiamate recuperano rispettivamente i file content/xml/article.xml e content/xml/forecast.xml per poi presentare i dati sulla pagina. Il database viene interrogato con query utilizzando la libreria PDO.