**Indice**

1. **Introduzione**
   1. I dati esistenti
   2. Proposta di soluzione
2. **Web services**
   1. Servizio UnimapServ
   2. Servizio su applicazione esami e questionari
   3. Servizio UgovRicerca
3. **Yii**
4. **Il progetto**
   1. Strumenti di base
   2. Header e footer
   3. Didattica
      1. Pagina dettaglio corso
   4. Curriculum
   5. Pubblicazioni
      1. Pagina dettaglio pubblicazione
   6. Index
      1. Contatti e ricevimento
5. **Layout**
   1. Bootstrap
      1. Caratteristiche
      2. Gryd system
   2. Stile
6. **Sviluppi** 
   1. Possibilità di personalizzazione
   2. Internazionalizzazione
7. **Conclusioni**
8. **Sitografia**
9. **Introduzione**

Per un’istituzione come l’Università di Pisa è importante fornire le proprie informazioni in maniera chiara, coerente ed accessibile; nel caso di docenti e ricercatori questo risulta ancora più vero, dato che è principalmente a loro che fa riferimento buona parte dell’attività dell’ateneo. Sarebbe quindi opportuno che esistesse una pagina per ciascuno di essi, contenente i loro dati ma anche indicazioni sui corsi e sulle attività, e più in generale ogni informazione ritenuta pertinente.

Sembra però che i docenti siano poco “supportati” nella creazione di una propria home page istituzionale, visto che anche ad una analisi superficiale ci si imbatte immediatamente in due grossi problemi, la mancanza di dati e la difficoltà nella loro organizzazione.

* 1. **I dati esistenti**

Visitando i siti dei vari dipartimenti, indicati sul sito dell’Università di Pisa alla pagina <https://www.unipi.it/index.php/strutture/itemlist/category/106>, si trova che su 18 siti di dipartimento solo 8 possiedono un elenco di pagine personali dei docenti, ma anche in questo caso si tratta spesso di link rotti, pagine vuote o pagine che riportano informazioni parziali: per citarne uno, il Dipartimento di Ingegneria dell’Informazione per ogni docente fornisce solo gli orari di ricevimento.

In 5 pagine di dipartimento è presente un elenco di nominativi collegati con la relativa pagina uniMap, mentre 3 riportano un elenco “misto”, in cui sono presente sia i nominativi collegati a uniMap sia, seppure solo talvolta, un collegamento esterno a siti personali dei docenti.

I due siti di dipartimento che non offrono agli utenti neppure una lista di nominativi sono quello del dipartimento di Farmacia, [www.farm.unipi.it](http://www.farm.unipi.it), e <http://www.med.unipi.it/>, che è indicato come il sito per tre dipartimenti differenti dell’area di Medicina; due pagine che dovrebbero quindi in teoria fornire informazioni a una assai ampia fetta di studenti.

Oltre agli elenchi delle pagine docente nei siti di dipartimento poi esistono quelli delle pagine dei singoli corsi, appartenenti principalmente dell’area di ingegneria, senza contare che talvolta esistono persino due siti differenti per la triennale e la magistrale di uno stesso corso . Esiste inoltre un elenco nella pagina della Scuola di Ingegneria, all’indirizzo <http://www2.ing.unipi.it/cgi-bin/homepages3>, circostanza che porta alla duplicazione di molte informazioni già presenti altrove, come pure il secondo elenco docenti nella sezione Didattica della pagina del Dipartimento di Biologia.

Inoltre gli elenchi stessi non sono affidabili, nella maggior parte dei casi non è presente la data di ultimo aggiornamento: per citare solo i siti di dipartimento, è riportata solo sotto l’elenco docenti del Dipartimento di Fisica, aggiornato al 2007, e quello del Dipartimento di Ingegneria civile e Industriale, aggiornata al 2014, che ha un elenco docenti di soli 2 elementi.

In tutte le home docenti è presente una varietà di dati che spazia da informazioni di contatto, come e-mail e telefono, che sembrano essere i soli dati ricorrenti (ma non sempre, come già visto), a materiali didattici, indicazioni su progetti di ricerca, siti personali esterni, e talvolta persino liste di hobby.

\*

Il risultato è fortemente disomogeneo e rende molto difficile, quando non praticamente impossibile, la consultazione di queste pagine. Anche quando si riesca a trovarle, la qualità dei dati in esse contenuta è spesso dubbia, quando non del tutto scadente, oppure è la strutture stessa della pagina che rende molto complicata la consultazione.

L’incredibile varietà di queste pagine è, fra gli altri, uno dei motivi per cui non sono riuscita a quantificare la ricorrenza di certe caratteristiche, ad esempio in quante di esse fossero presenti materiali didattici; ogni dipartimento ha una propria struttura, che in alcuni dipartimenti non consente modifiche e in altri sì, e ogni personalizzazione riporta ciò che è congeniale al proprietario della pagina. Non mi è stato quindi possibile neanche stabilire un criterio di valutazione di questi contenuti, e ho abbandonato l’idea di una ricerca in tal senso perché questi non sono validi nella maggior parte dei casi; risultava superfluo ai fini del progetto, perché, come illustrato nella sezione successiva, l’idea è quella di supplire alle mancanze presenti creando un nuovo tipo di pagina istituzionale, a partire da servizi esistenti che forniscono informazioni più valide, universali e più coerenti fra loro.

* 1. **Proposta di soluzione**

Negli ultimi anni sono stati sviluppati web services che, tra l’altro, rendono disponibili dati relativi ai singoli docenti. Sfruttando queste fonti si possono creare, senza neppure la necessità di intervento da parte dei docenti stessi, pagine istituzionali che forniscano informazioni affidabili; non esiste poi la necessità di replicare o aggiornare tali dati, perché vengono estratti al momento della richiesta della pagina da database che sono aggiornati periodicamente.

Inoltre, un progetto del genere permette di adottare un layout unico per tutte le pagine, rendendole più riconoscibili e più facili da consultare.

Se questo sistema fosse adottato dai vari dipartimenti, si avrebbe per ogni docente una pagine istituzionale unica, collocata nel sito del dipartimento a cui appartiene il docente, dove è possibile tra l’altro trovare informazioni su tutti i corsi e link a eventuali pagine personali esterne, eliminando il problema della ripetizione dei dati nei siti dei corsi di laurea.

1. **Web services**

I web service che ho utilizzato sono erogati dalla piattaforma WSO2 di ateneo, all’indirizzo IP <https://131.114.77.213:8263>. L’accesso avviene tramite autenticazione OAuth2(Bearer token), fornendo cioè il token assegnato al dipartimento di appartenenza, per il mio progetto quello del Dipartimento di Informatica. I servizi sono erogati sia in modalità SOAP che REST; la seconda modalità è quella utilizzata in questo caso, aggiungendo un header ”Accept:json” alla richiesta e specificando nella url i parametri necessari ai metodi invocati. La risposta ottenuta è un oggetto contenente altri oggetti e array associativi, i campi dei quali sono richiesti dai vari programmi in modo da ottenere le stringhe corrispondenti, inserite poi nei tag opportuni.

Il problema di molti di questi servizi è che talvolta le informazioni fornite sono parziali e non bastano a creare il contenuto richiesto in un certo ambito, quindi si rende spesso necessario estrarre codici identificativi dal primo service per utilizzarli come parametri nella chiamata a un secondo service per procurarsi i dati mancanti. Il tutto ovviamente allunga i tempi e rende disagevole navigare nei risultati alla ricerca del giusto valore. I casi più evidenti sono quelli di riguardanti le pubblicazioni e i corsi, che hanno molti dettagli suddivisi in array provenienti da più servizi (vedi sezione **Progetto**).

Inoltre anche le stringhe ottenute spesso non hanno la formattazione adatta, e vanno quindi ripulite e rese leggibili, ad esempio con la sostituzione dei simboli speciali, la normalizzazione dei nomi e l’inserimento di spazi, virgole e rientri a capo. Le sigle sono tradotte con l’uso di dizionari in forma di array associativi. Ovviamente anche le funzioni che si occupano di questo aspetto vanno a influire sulla rapidità del caricamento della pagina.

* 1. **Servizio UnimapServ**

“UniMap è il nuovo sistema di ricerca dei dati relativi al personale, alle strutture, all'attività didattica e all'organizzazione dell'Università di Pisa. I dati pubblicati su UniMap, aggiornati con cadenza giornaliera, provengono principalmente da due basi di dati: le informazioni relative al personale derivano dal database CSA, il sistema informatico di gestione giuridica ed economica del personale, mentre le informazioni relative all'attività didattica provengono dal database Esse3, il sistema informatico di gestione delle carriere degli studenti.   
UniMap è stato sviluppato con strumenti di pubblico dominio, come PHP, Oracle\_Lib, Smarty.”  (pagina iniziale uniMap)

Il servizio UnimapServ, con url <https://131.114.77.213:8263/unimapserv/1.0>, restituisce l'elenco del personale corredato da informazioni riguardanti il ruolo, la sede, eventuali cariche e un campo chiave per il link a uniMap, ricavato dal database uniMap.

Da questo web service provengono tutti i dati personali riguardanti il docente: nome e cognome, ruoli, cariche ricoperte, curriculum, mail, telefono e sito personale. Il metodo utilizzato è getPersona, che richiede il codice docente.

Il docente che volesse cambiare le informazioni sopra elencate deve effettuare la modifica direttamente dalla propria pagina uniMap; i cambiamenti si rifletteranno automaticamente sulla pagina istituzionale.

* 1. **Servizio su applicazione esami e questionari**

Il web service si trova all’indirizzo <https://131.114.77.213:8263/esami/1.0/>.

È stato utilizzato il metodo listaInsegnamentiDocente(aa, codice\_docente), che richiede l’anno dei corsi, indicato con quello del primo semestre dell’anno accademico, in questo caso “2016”, e il codice docente. Restituisce l’elenco degli insegnamenti di cui un docente fa parte, come titolare, responsabile di modulo o codocente.

Questo metodo è stato creato specificamente per questo progetto, sulla falsariga di listaInsegnamenti(cds, aa), che dato corso di studi e anno accademico restituisce l’elenco degli insegnamenti che corrispondono ai due parametri. I campi restituiti e il loro contenuto sono gli stessi per entrambi i metodi, con la differenza che i risultati di listaInsegnamentiDocente fanno riferimento ai corsi del singolo docente e non a quelli dell’intero corso di studi.

I dati richiesti a questo servizio sono insegnamento\_descrizione, a cui corrisponde la stringa che rappresenta il titolo dei corsi, e corso\_id, il codice identificativo del singolo corso che viene passato come parametro a dettaglioCorso.php: qui viene usato con i metodi listaModuli e programmaDettaglio per ottenere tutta una serie di informazioni che vanno a comporre una pagina dettagliata del corso corrispondente all’id fornito.(v. sezione **Didattica**).

* 1. **Servizio UgovRicerca**

Il servizio, che fa riferimento al database Ugov Ricerca e ha url https://131.114.77.213:8263/pubblicazioni/1.0, ha due metodi: getList richiede in ingresso la matricola del docente e restituisce l'elenco delle pubblicazioni del docente presenti in Ugov Ricerca, con il quale è creata la lista nella pagina Pubblicazioni, mentre getSingle richiede id\_prodotto, cioè il codice della pubblicazione (ricavato dal metodo precedente) della quale restituisce i dettagli che sono visibili visitando il link collegato al nome dell’opera nella suddetta pagina.

1. **Yii**

Per un primo tentativo di realizzazione della home docente è stato preso in considerazione Yii, un framework in PHP pensato per la realizzazione di applicazioni web e API.

Yii nasce nel 2008 da un tentativo del suo creatore, Qiang Xue, di risolvere gli inconvenienti di un precedente framework da lui creato, PRADO, in particolare la gestione lenta di pagine complesse, una curva di apprendimento molto ripida e la difficoltà nel personalizzare molti controlli. Il nome Yii in cinese significa *semplice ed evolutivo*, mentre in inglese è acronimo di **Yes, it is!**

“Is it fast? ... Is it secure? ... Is it professional? ... Is it right for my next project? ... **Yes, it is!” (www.yiiframework.com/about/)**

È basato sull’architettura MVC, che punta a separare la gestione del contenuto dalla gestione della visualizzazione. Il modello (M) rappresenta i dati, la logica e le regole dell’applicazione; la vista (V) è la rappresentazione dei dati tramite elementi dell’interfaccia utente come testo, form e tabelle; il controller (C) gestisce la comunicazione fra gli altri due elementi, traducendo l’input in comandi per l’uno o l’altro.

Yii risulta molto leggero e veloce ed è ampiamente personalizzabile. Ha tutta una serie di caratteristiche fra le quali:

- Strumenti per gestire dati provenienti da database come oggetti: [Database Access Objects (DAO), Query Builder, Active Record, DB Migration](http://www.yiiframework.com/doc/guide/database.overview).

- Schema di caching a strati, che supporta la memorizzazione nella cache di dati, caching di pagine e di frammenti, e contenuti dinamici.

- Generazione automatica di codice per lo scheletro dell’applicazione e per applicazioni CRUD tramite lo strumento Gii.

- Misure di sicurezza per evitare attacchi come SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), e cookie di manomissione.

- Buona integrazione con codice di terze parti e con jQuery

Nel 2014 è stata rilasciata la versione 2.0, una completa riscrittura rispetto alla versione precedente. L’attuale versione 2.0.9 supporta anche PHP7.

Nonostante queste notevoli caratteristiche, a una prova pratica l’utilizzo di questo framework si è rivelato troppo complesso rispetto al progetto che doveva essere realizzato. Yii è molto orientato alla gestione di dati provenienti da database ma, visto che i dati disponibili per il progetto provengono da web services forniti dall’Università di Pisa, risultava abbastanza dispendioso e inutile inserirli in un database per poi estrarli in un secondo momento. Dati del genere possono molto più facilmente essere gestiti da piccoli programmi in PHP, anche quando siano una grande quantità; nel caso delle pubblicazioni ad esempio è stata creata una copia dei dati in locale da richiamare all’occorrenza, aggiornandola periodicamente.

1. **Il progetto**

Per questo progetto l’idea è stata quella di creare una struttura “componibile” lato server, adattabile ai vari siti di dipartimento, che abbia tuttavia un’architettura di base nell’organizzazione dei menu e dei contenuti . La pagina del docente è realizzata da piccoli blocchi di programma in PHP che richiamano di volta in volta il web service più adatto, estraggono le informazioni richieste, le ripuliscono ove necessario e restituiscono stringhe o blocchi di HTML che possono essere assemblati per formare una singola pagina. Alcuni menu e contenuti sono quindi generati al volo con le informazioni del docente e inseriti laddove è richiesto.

Per il mio progetto ho utilizzato i dati della professoressa Simi, e ho preso a modello la sua pagina istituzionale, mantenendo il più possibile lo stile e la disposizione delle informazioni nella loro forma originale.

* 1. **Strumenti di base**

La chiavi necessarie per accedere ai web services, cioè il token di autenticazione $auth, e quella per richiedere i dati del docente, cioè la matricola $codice\_docente, sono memorizzate in **auth.php**, quindi per creare la pagina di un docente si devono innanzitutto impostare questi codici.

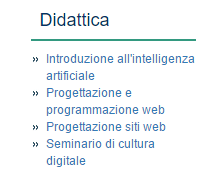
**utils.php** contiene gli strumenti principali per richiamare i vari web services, con la funzione getCurlJson, e trattare i dati che ne vengono ricavati , tramite funzioni utilizzate per ripulire e formattare le stringhe di testo in modo da renderle leggibili per l’utente (vedi sezioni seguenti).

* 1. **Header e Footer**

L’header attuale indica il Dipartimento di Informatica e il relativo menu, che ricalca quello della home di dipartimento, come pure il footer. Entrambi sono file contenenti codice HTML, il contenuto generato in questo caso è il nome del docente nel tag <title> e nel breadcrumb dell’header. I menu di sinistra sono anch’essi inclusi nel file header.php, dato che sono stati mantenuti in tutte le pagine per offrire uno strumento di navigazione semplice e riconoscibile ovunque all’utente (v. successivo) .

* 1. **Menu Didattica**

Il menu Didattica sfrutta come già detto il web service dell’applicazione esami e questionari; dai dati ottenuti sono estratte le stringhe corrispondenti al campo insegnamento\_descrizione, vale a dire il titolo del corso, che sono utilizzate per creare le voce del menu, ognuna un link alla pagina dettaglioCorso.php. Questa a sua volta è creata passando come parametro al file il codice del corso, corso\_id.

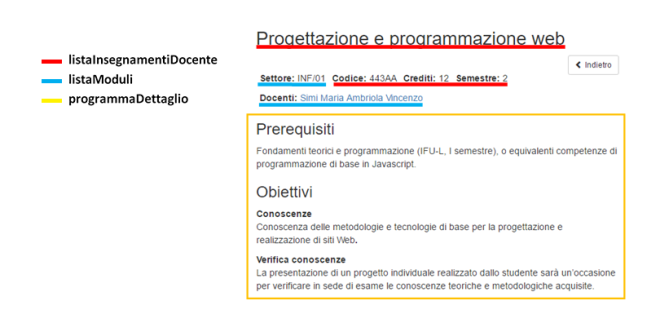
****

* + 1. **Pagina Dettaglio Corso**

La pagina è realizzata con informazioni provenienti da diversi metodi. Il programma in dettaglio corso.php richiama ancora una volta listaInsegnamentiDocente, con i parametri di matricola e anno, e scorre l’oggetto risultante per trovare l’array contenente i dettagli del corso con il codice corrispondente; di questi vengono quindi stampati titolo, lettera, codice, cfu e semestre del corso.

Dal metodo listaModuli, che richiede l’id del corso, provengono invece settore disciplinare e nome dei docenti in carica dei vari moduli, come pure chiave\_unimap\_docente, che è utilizzato per creare il link alla pagina uniMap di ciascun insegnante, disponibile all’indirizzo <http://unimap.unipi.it/cercapersone/dettaglio.php?ri>= (+ codice).

Infine il metodo programmaDettaglio, che richiede sempre l’id, restituisce l’array associativo contenente tutti i dettagli del programma come Prerequisiti, Conoscenze, Modalità ecc.; se questo risulta vuoto, viene stampato il messaggio *“Nessun programma disponibile”*.



* 1. **Curriculum**

Il curriculum è fornito dal servizio uniMap come un’unica lunga stringa, assieme a un eventuale link a una fonte esterna. Stranamente nello stesso array viene fornito anche l’orario di ricevimento, che forse sarebbe stato più utile inserire nell’array contenente tutte e altre informazioni di contatto. Il testo, altrimenti illeggibile, viene normalizzato e formattato, e la stringa del link è collegata all’indirizzo indicato.

* 1. **Pubblicazioni**

La pagina delle pubblicazioni è stata complessa da realizzare, come la pagina dettagliata dei singoli corsi, a causa del modo in cui sono strutturati i web services. Ho usato due metodi del servizio UgovRicerca: getList fornisce titolo della pubblicazione, anno, nome degli autori e codice della pubblicazione. Con quest’ultimo, passato come parametro al secondo metodo, è poi possibile estrarre ulteriori dettagli; infatti getList non fornisce informazioni quali casa editrice, volume e nome del libro contenenti la pubblicazione, che vengono invece richieste con getSingle. Per ogni pubblicazione è necessario estrarre queste informazioni, e ovviamente questo processo è molto dispendioso: ci possono volere anche più di 20 secondi per caricare la pagina. Per questo motivo la lista viene creata solo se non esiste in precedenza, inserendo il risultato del programma in un file che viene richiamato dalla pagina a tutte le seguenti richieste e viene aggiornato periodicamente.

Ovviamente affidando la creazione della bibliografia a uno script è impossibile controllare tutti i dettagli per verificare la presenza di eventuali errori nei dati o la mancanza di alcuni di essi; in questo caso mi sono limitata a inserire, se presenti, titolo, autori, nome dell’editore, volume e titolo dell’eventuale libro per ogni pubblicazione. Il contenuto del file è dunque una lista ordinata per anno delle pubblicazioni del docente; per ciascuna è disponibile una pagina dettagliata a cui si accede dal link nel titolo, che conduce a dettaglioPubblicazioni.php.

* + 1. **Pagina Dettaglio Pubblicazioni**

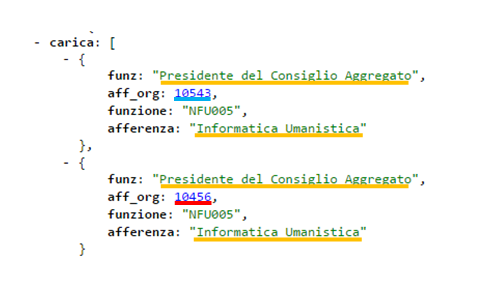
Utilizzando anche qui il risultato di getSingle con parametro l’id, si procede alla verifica del contenuto dei campi che, se non risulta vuoto, è stampato con la giusta indicazione. Il risultato è una pagina che mostra tutti i dettagli disponibili di una data pubblicazione.

* 1. **Index**

Nella pagina principale vengono mostrate le informazioni principali riguardo al docente, che corrispondono a quelle disponibili su uniMap. Nella colonna centrale sono elencati, oltre al nome e al cognome, i ruoli e le cariche ricoperte.

Nel caso del ruolo, l’informazione estratta corrisponde al codice relativo, che deve essere tradotto in modo da essere accessibile anche ai non addetti ai lavori; per questo ho creato un array associativo usato per sciogliere le varie sigle.

Riguardo alle cariche invece ho avuto difficoltà con i dati: nel caso specifico il codice identificativo delle due cariche era diverso perché la professoressa Simi è Presidente del Consiglio Aggregato di Informatica Umanistica sia per la triennale che per la magistrale, quindi funzione e afferenza risultavano le stesse, ma non c’era modo di indicare la differenza perché l’ indicazione del ciclo di laurea non è presente nei dati forniti dal web service. Si è quindi reso necessario eliminare la ripetizione che si verificava, confrontando funzione e afferenza per evitare di mostrare due stringhe riferite a cariche diverse sostanzialmente uguali.



* + 1. **Contatti e Ricevimento**

Il contenuto di questa colonna proviene da uniMap. Lo script controlla se i contatti non sono vuoti, poi li stampa. Per il ricevimento purtroppo il dato a disposizione è un’unica stringa, piuttosto lunga e difficilmente editabile.

In questo caso sarebbe preferibile avere un dato più strutturato, possibilmente contenente orario e indicazione estesa della sede di ricevimento, che indichi non solo il dipartimento ma anche l’indirizzo dello stesso e lo studio del docente. Sarebbe utile anche specificare le modalità di ricevimento, ad es. se è su appuntamento.

1. **Layout**

Tenendo conto dello scopo di questa pagina, ho cercato di organizzare il layout in modo che le informazioni fossero disposte nella maniera più chiara ed equilibrata possibile, dndo ad esmpio risalto ai contatti, ai quali è dedicata una colonna laterale, oppure assegnando più spazio al contenuto in altre pagine.

* 1. **Bootstrap**

Lo strumento che ho utilizzato per una migliore impaginazione è il framework gratuito Bootstrap.

Bootstrap è stato sviluppato da Mark Otto e Jacob Thornton presso [Twitter](https://it.wikipedia.org/wiki/Twitter) come un [framework](https://it.wikipedia.org/wiki/Framework_per_applicazioni_web) che uniformasse i vari componenti che ne realizzavano l'interfaccia web, dato che la presenza di diverse librerie aveva portato ad incoerenze ed elevati oneri di manutenzione. (…) Nell'agosto 2011 Twitter ha rilasciato Bootstrap come [open source](https://it.wikipedia.org/wiki/Open_source), invitando tutti gli sviluppatori a partecipare al progetto e a dare il loro contributo alla piattaforma. Nel febbraio 2012, è stato il progetto di sviluppo che sulla piattaforma [GitHub](https://it.wikipedia.org/wiki/GitHub) ha ricevuto il maggior numero di apprezzamenti, una posizione che detiene ancora nel giugno 2014 (…).

Bootstrap è uno dei framework frontend più popolari, usato per uno sviluppo web più veloce e più facile di siti web responsive e mobile-first.

Include modelli di design basati su HTML e CSS per tipografia, moduli, pulsanti, tabelle, navigazione, modali, caroselli di immagine e molti altri, nonché plugin JavaScript opzionale. Bootstrap è compatibile con tutti i browser moderni, (Chrome, Firefox, Opera, Safari e Internet Explorer). Risulta facile da utilizzare anche a chi abbia solo una conoscenza di base di HTML e CSS.

\*w3schools

* + 1. **Caratteristiche**

Bootstrap utilizza alcuni elementi HTML e proprietà CSS che richiedono l'utilizzo del doctype HTML5. Le classi dei tag sono utilizzate per indicare a quali regole di visualizzazione obbediscono gli elementi della pagina. I tag stessi obbediscono a regole tipografiche stabilite dal framework: i titoli <h1> hanno dimensione 36px, il testo 14 px. Tag e classi specifiche, come ad esempio <code> o .text-info, cambiano la grandezza, il colore e il tipo di visualizzazione del testo che contengono.

Il framework richiede un elemento contenitore per avvolgere i contenuti del sito e ospitare il sistema a griglia (v. successivo): i due tipi di contenitori disponibili, non annidabili, hanno classe .container, per un contenitore responsive a larghezza fissa, e .container-fluid, che invece crea un contenitore che si adatta completamente alla larghezza dello schermo, occupandone tutto lo spazio disponibile.

* + 1. **Grid System**

Bootstrap implementa un sistema a griglia usato per creare layout tramite una serie di righe e colonne che ospitano il contenuto. Le righe, con classe .row, sono contenute da .container o .container-fluid e creano gruppi di colonne orizzontali, dentro alle quali si trova il contenuto; uno spazio fra di esse è creato tramite padding. La pagina è suddivisa in 12 colonne, che possono essere raggruppate per formarne di più ampie. La classe utilizzata è .col-\*-\*: al primo asterisco corrisponde la larghezza dello schermo, che può essere sx per quello del cellulare, sm per quello del tablet, md per quello del desktop e lg per schermi più larghi; al secondo asterisco corrisponde invece il numero di colonne occupate dal contenuto del div. La somma delle colonne utilizzate in una riga deve essere sempre 12: ad esempio, se voglio separare il contenuto in tre colonne uguali, dovrò utilizzare come classe .col-sx-4 per ciascuna di esse. Se ci sono più di 12 colonne per riga, ogni gruppo di colonne in più è inserito in una nuova riga come una singola unità.

Sfruttando queste proprietà è possibile creare template per varie istanze di siti web come blog, social network e negozi online. Quello che ho utilizzato per il progetto è una forma modificata del template “Web Page” disponibile alla pagina <https://www.w3schools.com/bootstrap/bootstrap_templates.asp> .

In particolare, tranne che nella parte superiore dell’header, ho lasciato due colonne vuote a destra e a sinistra su ogni riga, dato che il contenuto della pagina non richiede molto spazio. Quindi la navbar che contiene il menu di dipartimento, la riga che contiene breadcrumb e language switcher, entrambi larghi 4 colonne, e il contenuto del footer, occupano tutti 8 colonne, con i menu presenti nel footer di larghezza 2 colonne ciascuno. Nella parte del contenuto, il menu di sinistra, presente in tutte le pagine, occupa anch’esso due colonne, e le restanti 6 sono dedicate di volta in volta alle diverse informazioni mostrate; l’unica eccezione è rappresentata dall’index, in cui il contenuto vero e proprio occupa le 4 colonne centrali, ed è affiancato da una colonna di larghezza 2 contenenti le informazioni di contatto.

\*

In ogni pagina diversa dall’index ho inserito un bottone che sfrutta diverse classi di Bootstrap e consente di tornare alla pagina precedente, per rendere più agevole la navigazione all’utente.

* 1. **Stile**

Bootstrap ha i propri stili prestabiliti, ma in questo caso era necessario creare una pagina che somigliasse il più possibile nell’aspetto a una pagina del sito del Dipartimento di Informatica; nel foglio stile.css sono elencate tutte le regole necessarie a questo scopo.

L’header e il footer hanno il gradiente blu dei siti dell’ateneo, e accanto al nome del dipartimento viene inserito il logo dell’Università di Pisa.

Nel menu principale è stato necessario cambiare il colore di sfondo, che Bootstrap rendeva scuro e non combaciava con quello del menu di dipartimento, ed è stato inserito il bordo inferiore grigio. L’header, la voce di menu principale attiva e i menu laterali hanno il bordo inferiore del verde #339980, che è il colore di Informatica.

Nelle liste principali per i punti è stata inserita l’immagine di una freccia singola, nelle liste secondarie quella di una freccia doppia e nel breadcrumb la freccia gialla, tipiche anch’esse del Dipartimento.

1. **Sviluppi**

Questo progetto è l’applicazione più semplice possibile per fornire le informazioni di base riguardo ogni docente. Ciò lascia ampi margini di miglioramento, e la possibilità di aggiungere alcune caratteristiche utili e interessanti.

* 1. **Personalizzazione**

Nel caso il docente ritenesse utile aggiungere personalmente informazioni alla propria pagina, la possibilità potrebbe essere garantita creando il sistema di inserirle e memorizzarle in un file da richiamare alle successive visite; ad esempio sarebbe utile creare uno spazio dove sia possibile inserire comunicazioni, informazioni dettagliate ed eventi afferenti all’attività didattica del docente, materiali didattici.

Il problema in questo caso sarebbe la creazione di un sistema identificativo per verificare l’identità dell’utente che intende apportare modifiche e reindirizzarlo alla propria pagina personale.

Sarebbe poi necessario stabilire il tipo di modifiche possibile, in modo da garantire una certa libertà ai docenti ma al tempo stesso evitare che una troppo fantasiosa personalizzazione renda difficile mantenere l’architettura della pagina, che in sostanza dovrebbe essere uguale per tutte le pagine istituzionali. Si dovrebbe innanzitutto stabilire cosa possa essere considerato utile e cosa invece sia superfluo per gli utenti finali, che nella maggior parte dei casi sono studenti: dispense e orari delle lezioni sicuramente sarebbero apprezzati, un elenco di hobby del docente potrebbe essere ritenuto irrilevante.

Ci sarebbe poi da stabilire in che forma inserire i dati; per alcune informazioni basta un’area di testo, mentre per informazioni più articolate, come ad esempio una lista di progetti di ricerca, sarebbe più adatto un menu.

Una differenza di cui tenere conto è quella fra le abilità informatiche dei docenti delle varie aree; un docente di Informatica non avrebbe problemi a gestire e personalizzare un proprio spazio web, mentre è probabile che i docenti dell’area umanistica non si trovino altrettanto a loro agio con tag e fogli di stile.

* 1. **Internazionalizzazione**

Altra modifica, piuttosto necessaria, è la creazione di uno script che generi la pagina in inglese. Al momento l’ostacolo sorge più che altro a livello dei dati provenienti da uniMap, dei quali non esiste la versione in inglese, mentre nei dati di pubblicazioni e didattica esistono già se non altro i campi corrispondenti, anche se resta da vedere quanti di essi effettivamente non siano vuoti.

1. **Conclusioni**

Come visto i dati dei docenti dell’ateneo pisano sono molto frammentati, incompleti e disomogenei, quando non del tutto mancanti. La mia tesi aveva l’obiettivo di fornire lo spunto per trovare una soluzione semplice a questo problema, creando una struttura di base a cui è possibile aggiungere una serie di miglioramenti, come illustrato nella sezione precedente.

La soluzione proposta è una struttura che si propone di essere leggera e veloce, i vari elementi della quale siano componibili e facili da modificare anche separatamente, rendendola così adattabile a tutti gli ambienti, in modo da poterla inserire anche in contesti diversi da quello in cui è nata. La pagina è stata creata a partire dalle informazioni disponibili, ed è sempre aggiornata perché basata su service esterni che forniscono dati affidabili. Il layout è stato pensato per mostrare nel modo più chiaro possibile i contenuti, suggerendo al tempo stesso l’appartenenza della pagina al sito del dipartimento, per non creare negli utenti quel senso di disorientamento che potrebbe crearsi visitando pagine tutte diverse dal sito che le ospita e fra loro.

Il progetto si basa sui dati che sono disponibili in questo momento, forniti da web service offerti dall’Università, quindi la soluzione proposta ha validità per lo stato attuale delle cose, e non tiene conto di eventuali necessità o richieste che potrebbero venire fuori da un confronto con gli utenti. In questo senso sarebbe utile un lavoro a monte sui dati a livello di dipartimenti o di ateneo, che oltre a stabilire quali potrebbero essere raccolti per arricchire l’offerta, comprenda una verifica sui dati già disponibili per valutarne il grado di completezza e la validità dei criteri con cui sono strutturati nei service; in tal modo si potrebbe fornire un servizio più completo e corretto, migliorando le informazioni già esistenti e aggiungendo ciò che risulta mancante.