**RELAZIONE D’ESECUZIONE DEL TIROCINIO**

L’obbiettivo dell’esperienza di tirocinio consiste nella creazione di un Sito sfruttando la piattaforma Liferay 7.1 completo di tema personalizzato, che dovrà ospitare contenuti di tipo Web Content costruiti con due tipi di paradigmi:

1. Modalità classica: definizione di Struttura, Modello e CSS dal tema configurato, usato per la creazione di contenuti.
2. Modalità nuova: definizione di Page Fragments, Page Templates e Content Pages per la creazione di contenuti dinamici.

L’intero progetto sarà portato a termine usando le best practice di stile aziendali e il versionamento remoto del codice usando Git, usando codice compatibile per i più comuni browser esistenti sul mercato.

Per realizzare il Sito saranno sfruttati i miglioramenti introdotti dalla suddetta versione di Liferay nel campo della usabilità, user experience, responsività rispetto ai dispositivi usati per la visualizzazione e di accessibilità per persone con disabilità motorio/visive.

Infine, sarà data una valutazione finale sulle differenze che sono venute a riscontrarsi fra le due modalità di creazione dei Web Content descritte sopra, dando un giudizio sull’utilità di entrambi i paradigmi e definendo i loro punti di forza in determinati scenari.

Il progetto è preventivato a durare per circa trenta giorni lavorativi ed è suddiviso in fasi distinte e separate:

* Training: per poter affrontare il progetto di tirocinio è necessaria acquisire una determinata preparazione improntata sulle tecnologie utilizzate dal Portale per la creazione, modifica e mantenimento della struttura dei Siti e ai loro componenti Portlet che forniranno il servizio di consultazione e gestione di Web Content.

I contenuti del training sono:

* 1. Corso di Amministratore di Portale Liferay 7.1;
  2. Corso di creazione e gestione dei Web Content;
  3. HTML, CSS, Javascript;
  4. SASS, FreeMarker, JQuery;
  5. Documentazione del tool Visual Studio Code e dell’ambiente per lo sviluppo di Temi personalizzati.
* Installazione dell’ambiente di sviluppo: sarà installato e utilizzato il bundle Liferay 7.1 CE, eseguendo la configurazione del suo server Tomcat;
* Creazione del Sito: attraverso l’interfaccia di Amministrazione Liferay sarà creato e configurato il Sito richiesto, inserendo Pagine complete di Portlet visualizzatori di Web Content appropriati;
* Creazione di Web Content di tipo Template: verrà creata una pagina adatta all’inserimento di Web Content strutturati in Template;
* Creazione di un tema personalizzato: sarà realizzato un tema ad-hoc per la pagina appena costruita usando i tool a disposizione e redigendolo in SCSS.
* Adattamento del tema secondo criteri di responsività: sarà applicata al tema della pagina appena creata la libreria Bootstrap per implementare tali criteri;
* Verifica della funzionalità: test e convalida del Tema Personalizzato;
* Creazione della pagina Fragment;
* Creazione della Navigation: attraverso un frammento ad-hoc, sarà importato il widget che implementa la navigation;
* Creazione di Web Content di tipo Fragment: verranno creati dei Web Content di prova secondo il paradigma nuovo e inseriti nei componenti portlet del Sito;
* Completamento del tema personalizzato: sarà realizzato un tema ad-hoc per la pagina appena costruita usando i tool a disposizione e redigendolo in SCSS.
* Adattamento del tema secondo criteri di responsività: sarà applicata al tema della pagina appena creata la libreria Bootstrap per implementare tali criteri;
* Validazione: sarà eseguita una validazione del progetto usando le specifiche di usabilità, responsività e accessibilità;
* Valutazione della creazione Web Content: saranno valutate le modalità di creazione dei Web Content secondo i due paradigmi, con la determinazione dei punti di forza e di debolezza di ciascun metodo.

Ripartizione del tempo di progetto in giorni lavorativi di otto ore ciascuno:

* Training

|  |  |  |
| --- | --- | --- |
| **ATTIVITA’** | **PREVENTIVO**  **(giorni)** | **CONSUNTIVO**  **(giorni)** |
| Corso di Amministratore di portale e contenuti web | 5 | 5 |
| HTML, CSS, JAVASCRIPT | 5 | 5 |
| SASS, Freemarker, JQuery | 5 | 5 |
| Documentazione, best practices e Setup dell’Ambiente di sviluppo | 2 | 2 |

* Realizzazione pagina con Web Content di tipo template

|  |  |  |
| --- | --- | --- |
| **ATTIVITA’** | **PREVENTIVO**  **(giorni)** | **CONSUNTIVO**  **(giorni)** |
| Costruzione della struttura della pagina con Web Content statici | 2 | 2 |
| Realizzazione tema della pagina con Web Content di tipo Template | 2 | 2 |
| Verifica aderenza delle specifiche della struttura del Sito | 1 | 1 |
| Adattamento del tema ai criteri aggiuntivi di responsività riscontrati con Bootstrap e altre modifiche riscontrate nella fase precedente | 2 | 2 |

* Realizzazione pagina di Web Content di tipo Fragment

|  |  |  |
| --- | --- | --- |
| **ATTIVITA’** | **PREVENTIVO**  **(giorni)** | **CONSUNTIVO**  **(giorni)** |
| Training sui Fragment | 1 | 1 |
| Costruzione dei Web Content di tipo Fragment | 2 | 2 |

* Fase finale

|  |  |  |
| --- | --- | --- |
| **ATTIVITA’** | **PREVENTIVO**  **(giorni)** | **CONSUNTIVO**  **(giorni)** |
| Validazione progetto secondo le specifiche di usability, user experience, accessibility e responsivity | 2 | 2 |
| Documentazione differenze fra creazione contenuti semplici e dinamici | 3 | 3 |

Totale giorni preventivati: 32

Totale giorni consuntivati: 32

**ESPERIENZA DI REALIZZAZIONE**

La prima fase del progetto si è focalizzata nell’acquisizione delle conoscenze necessarie a manipolare l’ambiente di sviluppo affrontando il corso di Amministratore di portale Liferay, imparando le diverse modalità di creazione, modifica e gestione dei siti attraverso il portale.

Successivamente, si è resa necessaria la formazione improntata sui linguaggi di programmazione usati per la creazione del sito, HTML, CSS, Javascript, e sulle loro estensioni, FreeMarker, SASS e JQuery.

Dopo aver preso confidenza con tali linguaggi, è stato preparato l’ambiente di sviluppo installando la versione 7.1 del bundle di Liferay e configurandolo correttamente, impostando le variabili locali necessarie per il suo funzionamento. Dopo aver installato l’editor Atom e configurando l’ambiente di sviluppo dei temi installando liferay-theme-manager e il build manager Gulp, si è avviata la copia del server Tomcat contenuta nel bundle, configurata l’istanza di Liferay e si è provveduto alla creazione del sito contenente una singola pagina con contenuti Web Content di tipo Template.

La pagina è stata costruita prendendo come ispirazione la pagina iniziale del sito web nvidia.com, tentando di imitarne le funzionalità e stile ma utilizzando le tecnologie Liferay. Si sono create due strutture, una per l’inserimento di una sequenza di immagini dotate di titolo e descrizione, poi una seconda che estende la prima, aggiungendo un titolo generale alla stessa.

Dopo aver creato le strutture, si sono da esse create i template di tipo Banner e Squared Display per implementare rispettivamente un banner di tipo carosello e un display di contenuti a schede quadrate con titolo in sovraimpressione.

Dopodiché, si è provveduto a creare un tema che modificasse lo stile degli elementi dei template per avvicinarsi allo stile del sito preso come modello. In particolare, per creare il Banner si è usata la libreria esterna Slick.js e lo stile di display Flexbox per il contenitore a schede rettangolari.

Fatto ciò, si sono aggiunti alla pagina una portlet di tipo Navigation, un template di tipo Banner e tre template di tipo Squared Display, completando la prima versione della pagina.

All’ispezione del risultato si è reso necessaria la modifica del tema per incrementare la responsività includendo le definizioni di Bootstrap.js e delle media query per incrementare la capacità d’adattamento dei componenti della pagina a dispositivi con viewport ridotto.

Infine, si è proceduto con la modifica del codice in modo da rispettare gli standard aziendali per quanto riguarda i nomi delle variabili e per modificare ulteriormente lo stile di visualizzazione delle portlet, in modo da rimuovere le parti della struttura della pagina non utili come il titolo della portlet, che non risulta removibile in modalità barebone per un bug di Liferay.

Dopo una verifica della responsività, si è reso necessaria la modifica delle classi di stile del tema per utilizzare la tecnologia Bootstrap in modo da semplificare il codice di styling e migliorando significativamente la capacità di adattamento della pagina a diversi viewport.

Per terminare la prima fase del progetto, si è reso necessaria una riorganizzazione del codice e l’introduzione del versionamento su Git.

La seconda parte del progetto è costituita dall’acquisizione delle conoscenze necessarie per costruire una pagina Web di tipo Fragment Page e la realizzazione della pagina strutturalmente simile al Widget Page e confrontarne la modalità di creazione, la flessibilità dei componenti e dei contenuti inseribili.

Un Fragment è un pezzo di pagina HTML con associato codice CSS/JS inserita inline in un contenitore padre che si estende verticalmente sulla pagina. Lo sviluppo di un Fragment consiste nello specificare il suo codice HTML, CSS e JavaScript includendo dei tag specifici per inserire contenuti direttamente nel frammento deployato in modalità edit, sostituendoli a dei placeholder, con immagini e testo forniti dal Content Manager e a mano rispettivamente.

Avendo preso in considerazione tali informazioni, si è capito che i contenuti Freemarker delle Widget Applications precedentemente sviluppate sono riusabili così come tutte le regole di stile definite per il tema della pagina precedente. In ogni Template si è prelevato il codice Freemarker associato, convertito in HTML e lo si ha innestato in un Frammento e, partendo da essa, innestando i tag immagine e titolo assieme ai tag lfr-editable di tipo image e text rispettivamente.

Per le regole di stile, è stato deciso di mantenere le classi usate precedentemente definite ed ampliarle dove necessario, poiché le classi della pagina di supporto ai template non usano esattamente le stesse classi e id della pagina Template.

Inseriti i frammenti creati nella pagina, si sono poi modificati con la modalità edit per inserire i contenuti del Banner con carosello, i titoli e le immagini delle cards così da prendere una pagina molto simile alla precedente creata con i Widget.

Realizzata la pagina, è stata rivisto il progetto e compilata la relazione con le differenze fra lo sviluppo di una pagina Template e di una pagina Fragment.

**DIFFERENZE FRA TEMPLATE E FRAGMENT**

1. **Layout di pagina fisso**  
   Il contenuto inserito con i Fragment non è organizzabile in layout a colonne o righe multiple, solo in un'unica colonna. Ciò limita molto la capacità del programmatore di costruire pagine con complessi layout poiché la natura inline dei frammenti lo impedisce.
2. **Strutturazione dei dati**

I dati di un frammento non sono incapsulati in una struttura ma sono selezionati dal Media Manager del Sito, oppure sono inseriti a mano dalla funzione Edit nel caso di testo. Questo permette di organizzare più efficacemente le risorse in modo da forzare l’utente a creare una gerarchia dei contenuti nel Content Manager, che poi saranno condivisi ordinatamente da tutto il sito.

1. **Modifica veloce della pagina**

Rispetto alle pagine create con i Template, i Fragment sono velocemente modificabili senza operare su Strutture ma agendo direttamente sul contenuto visualizzato in pagina. Perciò pagine statiche sono modificabili molto più velocemente rispetto ad una struttura.

1. **Stile e codice incorporato**

I Fragment incapsulano le direttive dello stile e le funzioni Javascript direttamente al loro interno in modo da poter definire stili personalizzati per ogni tipo di frammento. Però questa funzionalità è pesantemente limitata dalla mancanza delle variabili, nesting e short-hand di SCSS/JQuery rendendo il codice meno pulito, più prolisso e più difficile da manipolare. La definizione di tali funzionalità attraverso il tema sarebbe da preferire anche per il fatto che tali regole vengono applicate dopo il caricamento della pagina fornendo bassa esperienza di navigazione. Inoltre, Bootstrap è utilizzabile anche per i Fragment.

1. **Natura statica della struttura della pagina**

I Fragment non sono costruiti per adattarsi dinamicamente al contenuto che devono rappresentare, cioè per aggiungere o togliere i loro sotto-elementi visivi è necessario manipolare direttamente il codice HTML, mentre per i contenuti Template è possibile scrivere direttive Freemarker che costruiscono pagine complessità arbitraria manipolando solo la struttura dati sottostante. La struttura di un contenuto di tipo Template è modificabile da una redazione, mentre quello del Fragment è solo operabile attraverso personale specializzato.

1. **Restrizione sul tipo di contenuti**

Il tipo di contenuto modificabile in un frammento è molto limitato: il tag lfr-editable di attributo type settato a “text” può modificare solo l’InnerHTML di un tag H, oppure se ha l’attributo type settato a “image” può solamente modificare l’attributo href di un tag IMG. Tutto ciò che non rientra nelle categorie sopra non è modificabile, come i link, visto che altri tipi di attributi non sono modificabili. Questo porterebbe a costruire funzioni Javascript per sopperire alla loro mancanza rendendo questa soluzione molto poco elegante.

1. **Supporto molto limitato ai Widget**

E’ possibile inserire Widget come elementi inline nelle pagine Fragment, ibridando la pagina. Sono solamente disponibili i Widget instanziabili forniti di default dalla piattaforma Liferay e la modifica del loro markup non è possibile, tranne per il loro stile, che però viene applicato solo al load della pagina intera provocando fastidiose storture durante il caricamento della stessa.

1. **Import/Export dei Fragment**

I Fragment supportano l’esportazione ed importazione dei Fragment in modo da permettere la condivisione e il backup più facile del codice. I Template e le Strutture non hanno tale funzionalità e solo la copia manuale del codice garantisce l’applicabilità del versionamento.

Con tale analisi, si conclude l’esperienza di tirocinio.