|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Progetto Outlook  Tesi di maturità |
| |  |  |  | | --- | --- | --- | | Daniele Del Giudice | A.S. 2012/2013 | 5A INA | |

Indice

[abbreviazioni utilizzate 4](#_Toc358998358)

[Introduzione 5](#_Toc358998359)

[Perché nasce il progetto 5](#_Toc358998360)

[Il progetto 5](#_Toc358998361)

[Descrizione del progetto 5](#_Toc358998362)

[Come si è svolta la realizzazione 5](#_Toc358998363)

[Strumenti 5](#_Toc358998364)

[Microsoft Access 5](#_Toc358998365)

[Sublime Text 2 5](#_Toc358998366)

[XAMPP 5](#_Toc358998367)

[Git + Github 5](#_Toc358998368)

[Convenzioni utilizzate 5](#_Toc358998369)

[Creazione di una libreria 5](#_Toc358998370)

[Utilizzo della lingua inglese 5](#_Toc358998371)

[Analisi del progetto 6](#_Toc358998372)

[Struttura logica del database 6](#_Toc358998373)

[Analisi entità 6](#_Toc358998374)

[Analisi relazioni tra le entità 7](#_Toc358998375)

[Struttura fisica del database 7](#_Toc358998376)

[La tabella given\_answers 8](#_Toc358998377)

[Flessibilità dell modello utilizzato 8](#_Toc358998378)

[Problematiche delle dipendenze domanda – risposta 9](#_Toc358998379)

[Normalizzazione 9](#_Toc358998380)

[Tecnologie utilizzate 10](#_Toc358998381)

[PHP, MySQL e MySQLi 10](#_Toc358998382)

[HTML5 e CSS3 10](#_Toc358998383)

[Javascript e jQuery 11](#_Toc358998384)

[Parti di codice php (submit delle risposte) 12](#_Toc358998385)

[Parti di codice js (mostra/nascondi domande) 12](#_Toc358998386)

[Considerazioni sulla sicurezza 12](#_Toc358998387)

[Transazioni 12](#_Toc358998388)

[Prepared Statements 12](#_Toc358998389)

[“Sicurezza delle password” 12](#_Toc358998390)

[Ulteriori considerazioni 12](#_Toc358998391)

[Realizzazione DNS 12](#_Toc358998392)

[Conclusioni 12](#_Toc358998393)

[APPENDICE 13](#_Toc358998394)

[Codice sorgente del progetto 13](#_Toc358998395)

[Bibliografia 13](#_Toc358998396)

# abbreviazioni utilizzate

# Introduzione

## Perché nasce il progetto

# Il progetto

## Descrizione del progetto

# Come si è svolta la realizzazione

## Strumenti

### Microsoft Access

### Sublime Text 2

### XAMPP

### Git + Github

## Convenzioni utilizzate

### Creazione di una libreria

### Utilizzo della lingua inglese

# Analisi del progetto

## Struttura logica del database

Il nucleo centrale del servizio realizzato risiede all’interno del database. Esso infatti contiene tutti i dati relativi agli studenti, al questionario, e alle risposte date da ciascuno studente.

Il modello scelto per la realizzazione è di tipo relazionale. Si procede ad illustrare la sua struttura logica, attraverso l’utilizzo di un diagramma Entity-Relationship:



Figura 1 – Diagramma Entity-Relationship del modello scelto

### Analisi entità

Le entità identificate nella modellazione del database sono le seguenti:

* sections – contiene un elenco delle sezioni (gruppi di domande relative ad un argomento comune)
* questions – contiene un elenco delle domande presenti all’interno del questionario
* question\_types – contiene un elenco delle varie tipologie di domande presentate
* answers – contiene un elenco di tutte le possibili risposte per ciascuna domanda
* users – contiene un elenco degli studenti a cui verrà presentato il dizionario
* specializations – contiene un elenco delle singole specializzazioni di cui gli studenti hanno fatto parte

### Analisi relazioni tra le entità

Le relazioni presenti tra le entità identificate, sono le seguenti:

* sections - questions (1/N) – ciascuna sezione contiene una o più domande
* question\_types - questions (1/N) – ciascuna domanda appartiene ad una delle tipologie stabilite
* questions - answers (1/N) – a ciascuna domanda sono associate un certo numero di risposte
* answers - users (M/N) – un singolo studente può scegliere più risposte all’interno del questionario, ma allo stesso modo una singola risposta può essere scelta da più studenti distinti
* specializations - users (1/N) – ciascuno studente è stato iscritto ad una determinata specializzazione durante il suo corso di studi

## Struttura fisica del database



Figura 2 – Diagramma UML della struttura fisica del database

Si illustra in figura 2 la struttura fisica scelta per l’implementazione della base di dati, realizzata mediante l’RDBMS (Relational Database Management System) MySQL.

### La tabella given\_answers



Figura 3 – Particolare del diagramma UML della struttura del database, che mostra la tabella given\_answer

Un limite dei database relazionali consiste nella loro impossibilità di realizzare relazioni many-to-many (M:N), come quella presente tra le entità users e answers. Tale limite è però facilmente superabile mediante l’introduzione di una tabella di giunzione (junction table), e a due relazioni di tipo one-to-many (1:N) dalle due tabelle coinvolte a quest’ultima.

La junction table realizzata assume il nome given\_answer, e contiene un elenco delle risposte date da ciascun utente. E’ opportuno notare come tecnica permetta la semplice risoluzione di problemi altrimenti ostici, come la scelta di risposte multiple per una singola domanda (è sufficiente inserire più entry, una relativa ad ogni risposta).

### Flessibilità dell modello utilizzato

Una delle maggiori sfide presenti all’interno del progetto è la realizzazione di una base di dati flessibile, capace di adattarsi in modo rapido e semplice ad eventuali cambiamenti. La struttura proposta segue tale principio, in quanto ogni singola informazione relativa al questionario è compresa all’interno del database.

Aggiungere una nuova domanda, cambiarne la tipologia o l’ordine, correggere il testo di una risposta: sono tutte operazioni effettuabili mediante semplici modifiche all’interno del database. Non è richiesta alcun tipo di modifica al codice dell’applicazione.

Questo permette di ridurre gran parte dei costi di manutenzione, e limita l’eventualità di compromettere il sistema nel caso siano necessarie modifiche.

### Problematiche delle dipendenze domanda – risposta



Figura 4 – Particolare del diagramma UML della struttura del database, che mostra le tabelle questions e answers

Nell’ottica di una completa separazione tra dati e applicativo, risulta necessario introdurre un meccanismo per modellare le dipendenze presenti tra determinate domande ed altre risposte. Si consideri, per esempio, come ad uno studente che ha dichiarato di non aver mai frequentato l’università, non sia necessario effettuare domande relative al corso di laurea scelto.

La soluzione scelta consiste nell’introduzione di un campo opzionale dependency (dipendenza) all’interno della tabella questions. Se la risposta identificata da tale campo è stata selezionata in precedenza la relativa domanda viene mostrata. Se tale risposta non è stata selezionata, la domanda invece non viene mostrata.

### Normalizzazione

Nell’ottica dell’eliminazione di ridondanze informative, e conseguentemente del rischio introdurre uno stato di incoerenza all’interno del database, sono stati seguiti una serie di accorgimenti volti alla riduzione dello stesso alla terza forma normale.

La normalizzazione consiste di fatto nella riduzione di tabelle che presentano campi interdipendenti in tabelle più piccole. Nel caso sia necessario ottenere la tabella originaria per effettuare un’interrogazione è possibile utilizzare un’operazione di join.

## Tecnologie utilizzate

Si effettua adesso una rassegna delle tecnologie impiegate. In una successiva sezione si avrà modo di analizzare nello specifico l’utilizzo di ciascuna tecnologia all’interno del progetto Outlook.

### PHP, MySQL e MySQLi

Una tecnologia fondamentale ai fini della realizzazione dell’applicativo lato server è PHP.

PHP,acronimo di “PHP: Hypertext Preprocessor”, è un linguaggio di programmazione interpretato nato per la programmazione web. Al giorno d’oggi è largamente utilizzato per sviluppare applicazioni *lato server*.

All’interno dell’applicativo PHP (nello specifico la versione 5.2.17) viene utilizzato per la creazione di pagine dinamiche, come quella di presentazione del questionario, e per la realizzazione dei meccanismi di login e di invio delle risposte.

L’interfaccia di collegamento tra l’interprete PHP e l’RDBMS MySQL è fornita dalla libreria MySQLi (MySQL Improved). Tale libreria consiste essenzialmente in un miglioramento della precedente libreria MySQL: permette un approccio di tipo orientato agli oggetti (mantenendo intatta la possibilità di sfruttare un approccio procedurale), introduce l’utilizzo dei prepared statements (fondamentali per eliminare il rischio di attacchi di tipo SQL Injection) e permette la realizzazione di tecniche quali le transazioni.

### HTML5 e CSS3

La realizzazione del sito web collegato al servizio Outlook, è stata realizzata mediante l’utilizzo dei linguaggi di markup HTML5 e CSS3.

HTML5 (Hyper Text Transfer Protocol) è comunemente utilizzato per la realizzazione di pagine web.

CSS3 (Cascading Style Sheets) è usato per definire la formattazione e lo stile di documenti HTML.

L’utilizzo di tali tecnologie permette una completa separazione tra il significato semantico del contenuto e l’aspetto grafico, relativo al modo in cui i contenuti sono presentati. I vantaggi sono una maggiore chiarezza e la possibilità di riutilizzare parti di codice.

### Javascript e jQuery

La natura dinamica del sito web realizzata, è possibile grazie all’utilizzo di tecnologie quali il Javascript, e di una sua potente libreria, jQuery.

JavaScript è un linguaggio di scripting orientato agli oggetti comunemente usato nella creazione di siti web, lato client-side. Tutti i moderni browser web, infatti, dispongono di un interprete Javascript.

jQuery è una potente libreria Javascript che si propone lo scopo di semplificare la programmazione lato client delle pagine HTML. Per farlo, fornisce una serie di strumenti che permettono di effettuare svariate operazioni con poche e semplici linee di codice.

L’utilizzo di queste tecnologie ha trovato uso, specificatamente, nella realizzazione del meccanismo di visualizzazione di domande dipendenti da determinate risposte.

## Considerazioni sulla sicurezza

Ogni applicativo, prima di essere rilasciato, deve necessariamente affrontare una fase di revisione e di considerazioni su quanto riguarda la sicurezza e l’affidabilità dello stesso. Si esamineranno adesso alcuni aspetti fondamentali; in una sezione successiva si vedrà in che modo è stata eseguita l’implementazione vera e propria.

### Prepared Statements

Applicazioni web, che devono necessariamente interfacciarsi con un grande numero di persone, sono esposti ad una serie di attacchi che minano l’integrità dei dati memorizzati o dei servizi stessi. Una tecnica di attacco tra le più semplici da realizzare, ma non per questo meno pericolosa, è quella delle SQL Injection.

### Transazioni

In un database accessibile contemporaneamente da più utenti, come quello utilizzato da questo applicativo, assume grande importanza l’utilizzo delle transazioni per il mantenimento dell’integrità dei dati memorizzati. Una transazione consiste in una sequenza di operazioni eseguite in modo atomico, che comporta il passaggio del database da uno stato iniziale consistente Si ad uno stato consistente Sn.

Le transazioni hanno due principali funzioni:

1. Isolamento dei processi di accesso e di modifica eseguiti da più utenti in modo concorrente: se non si garantisce l’esecuzione di ciascuna transazione in modo atomico, i risultati ottenuti possono essere inesatti.
2. Mantenimento del database in uno stato di coerenza: se anche una sola delle operazioni contenute in una transazione fallisce, il database viene riportato allo stato iniziale Si.

È possibile riassumere le quattro proprietà fondamentali che che devono soddisfare i DBMS che implementano le transazioni, perché queste operino in modo corretto sui dati, mediante l’acronimo ACID (Atomicity, Consistency, Isolation, Durability).

È possibile applicare tale tecnica in tutti i casi in cui dati inseriti dall’utente vengono utilizzati per costruire dinamicamente una query SQL. Dei dati opportunamente costruiti, infatti, forniscono al malintenzionato la capacità di leggere e modificare i dati presenti nel database.

# Il sito web

Il sito web rappresenta il vero e proprio centro dell’applicazione. Attraverso di esso, infatti, gli ex-studenti sono in grado di compilare il questionario.

Collegandosi al sito del servizio Outlook viene presentata una pagina d’introduzione. In seguito si accede alla pagina di login. Come indicato in precedenza a ciascun ex-studente è stato fornito una password che lo identifica, e permette l’accesso al servizio.

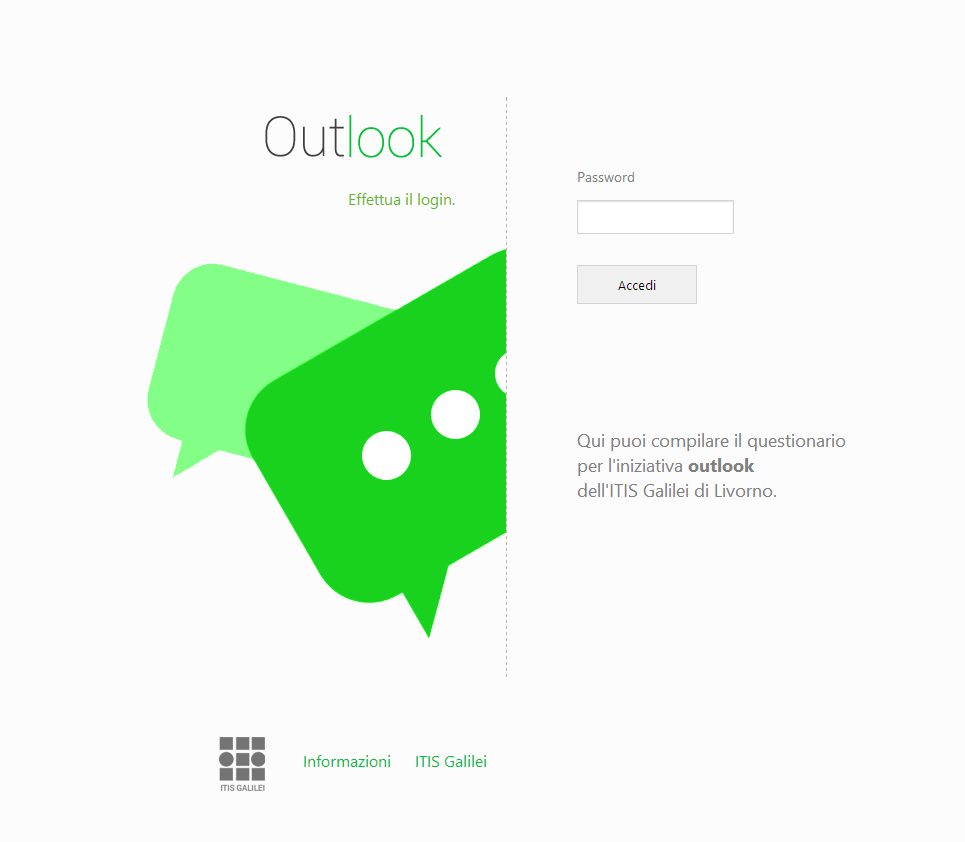


Figura – Pagina di login del sito web

Inserendo una password valida viene presentata la schermata contenente le domande del questionario. Nella parte superiore sono mostrati alcuni dati relativi all’ex-studente che sta compilando il questionario, mentre nella inferiore è presente il questionario vero e proprio. In fondo alla pagina sono presenti i crediti, dei collegamenti al sito della scuola ed un link che può essere utilizzato per effettuare il logout.

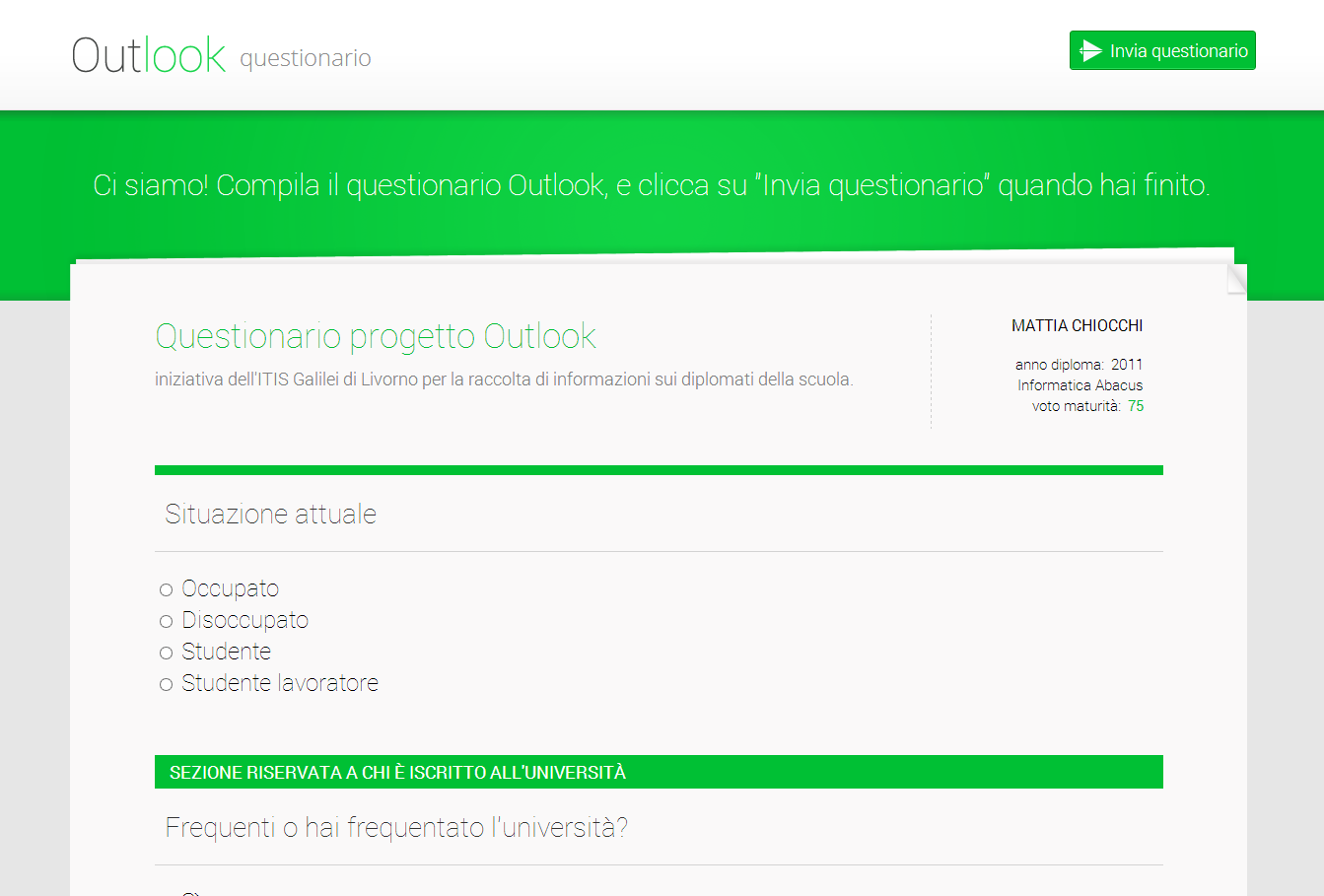


Figura – Pagina del sito web contenente il questionario

Il questionario è composto da varie sezioni, e parte delle domande vengono mostrate (o nascoste) in base alle risposte date alle domande precedenti. Si procede a titolo illustrativo a mostrare cosa accade quando si risponde positivamente alla domanda “Frequenti o hai frequentato l’università?”.

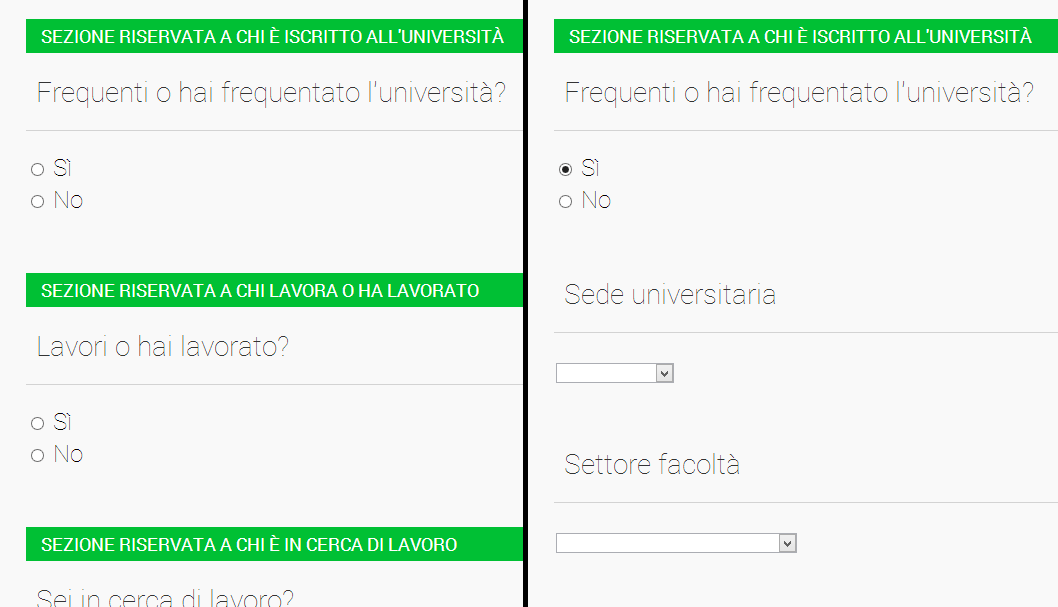


Figura – Particolare di domanda condizionata da una risposta

Terminata la compilazione del questionario è possibile effettuare l’invio delle risposte date. In seguito alla pressione del pulsante relativo (posizionato nella parte alta della pagina) viene effettuato un controllo lato-client per verificare che ogni domanda abbia effettivamente ottenuto una risposta. In seguito le risposte vengono inviate al server, che provvede a memorizzarle.

Se tutto è andato a buon fine viene presentata una pagina di avvenuto completamento, tramite la quale è possibile raggiungere il sito della scuola o tornare al questionario. Se l’utente ha già completato l’inserimento, infatti, vengono mostrate le risposte inserite in precedenza, e non risulta possibile inviare nuovamente il questionario.

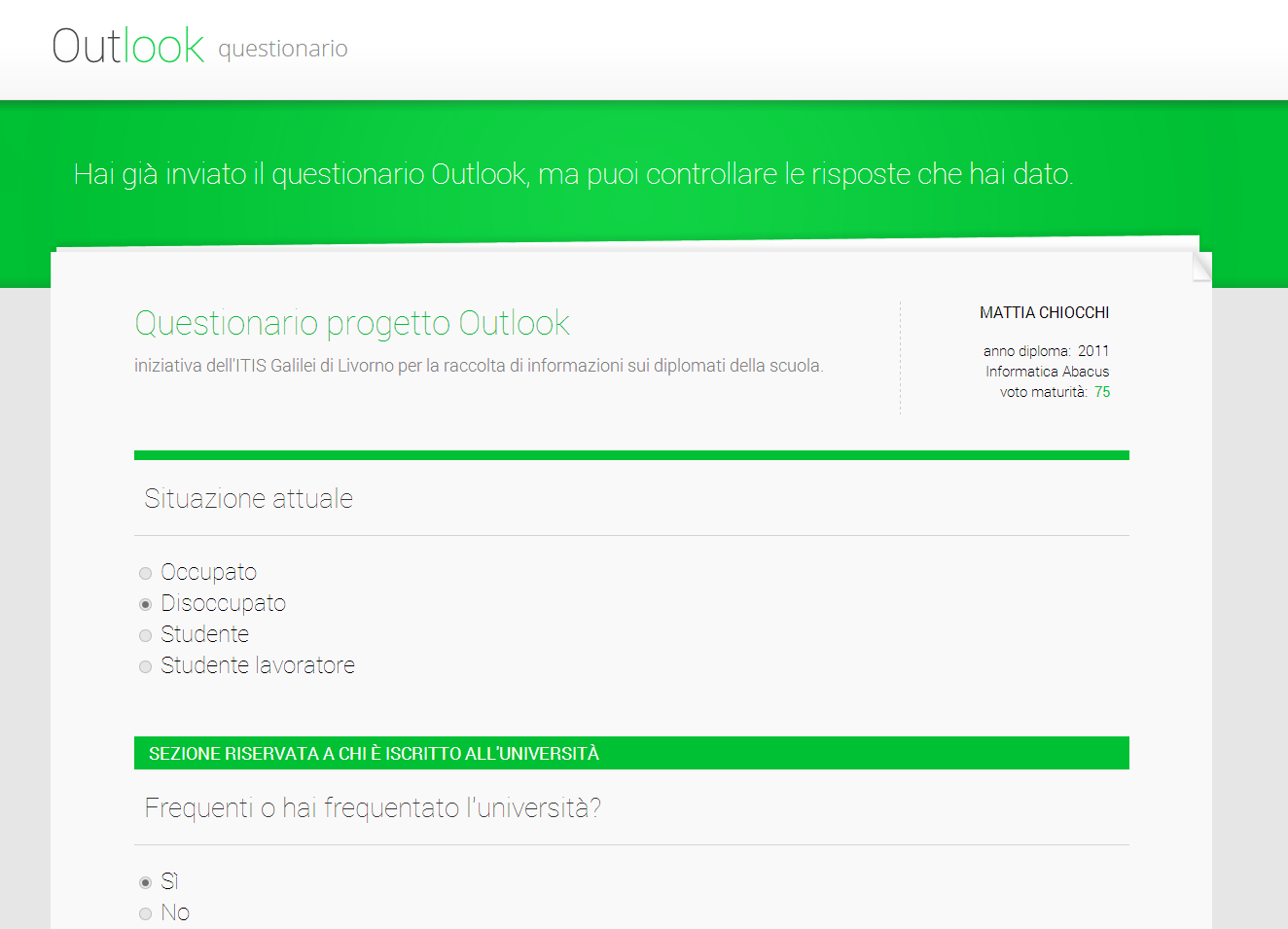


Figura – Pagina del questionario dopo averlo inviato

## Parti salienti di codice php

## Parti di codice js (mostra/nascondi domande)

## Ulteriori considerazioni

### Realizzazione DNS

# Conclusioni

# APPENDICE

## Codice sorgente del progetto

L’intero progetto è rilasciato sotto MIT License, è possibile scaricare il codice sorgente:

* Tramite Git: <https://github.com/D4n13le/outlook.git>
* Tramite link diretto: <https://github.com/D4n13le/outlook/archive/master.zip>

## Bibliografia

1. Formichi, Meini – Corso di Informatica Volume 3 – Zanichelli, 2012
2. Formichi, Meini – Il Linguaggio PHP – Zanichelli – 2012
3. Gigliotti – HTML 4.0.1 – Apogeo – 2004
4. Bigatti, Picca, Noel Castro – Sistemi di gestione di basi di dati e SQL– Apogeo – 2007
5. Venuti – JavaScript dalle basi ad AJAX – Edizioni Fag – 2008
6. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification – <http://www.w3.org/TR/CSS21/> – 20/05/2013
7. HTML5, a vocabulary and associated APIs for HTML and XHTML – <http://www.w3.org/TR/html5/> – 20/05/2013