



# Test Plan

**Tech4All**

Riferimento	C06_TP_1.0
Versione	1.0
Data	15/01/2025
Destinatario	Top Management
Presentato da	Marco Capuano, Giovanni Cerchia, Arcangelo Ciaramella, Silvana De Martino, Giovanni Esposito, Luigi Nasta, Giovanni Salsano, Giuseppe Staiano
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
26/12/2024	0.1	Aggiunta Introduzione.	Tutti.
26/12/2024	0.2	Aggiunta Relazione con altri documenti.	Tutti.
26/12/2024	0.3	Aggiunta Approccio.	Tutti.
26/12/2024	0.4	Aggiunta Panoramica del Sistema.	Tutti.
27/12/2024	0.5	Aggiunte Features da Testare.	Tutti.
27/12/2024	0.6	Aggiunti Pass/Fail Criteria.	Tutti.
28/12/2024	0.7	Aggiunto Sospensione e Ripresa.	Tutti.
29/01/2025	0.8	Aggiunto Test Case	Tutti.
08/01/2025	0.9.1	Revisione Materiale di Testing.	Luigi Nasta, Silvana De Martino.
08/01/2025	0.9.2	Revisione Test Case e Feature da testare	Arcangelo Ciaramella, Marco Capuano
15/01/2025	1	Revisione Test Plan.	Giovanni Salsano, Giuseppe Staiano.



## Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Ferdinando Boccia	Project Manager	FB	f.boccia28@studenti.unisa.it
Domenico D'Antuono	Project Manager	DD	d.dantuono7@studenti.unisa.it
Silvana De Martino	Team Member	SDM	s.demartino30@studenti.unisa.it
Luigi Nasta	Team Member	LN	l.nasta4@studenti.unisa.it
Giovanni Salsano	Team Member	GSA	g.salsano14@studenti.unisa.it
Arcangelo Ciaramella	Team Member	AC	a.ciaramella7@studenti.unisa.it
Giovanni Esposito	Team Member	GE	g.esposito282@studenti.unisa.it
Giovanni Cerchia	Team Member	GC	g.cerchia6@studenti.unisa.it
Marco Capuano	Team Member	MC	m.capuano37@studenti.unisa.it
Giuseppe Staiano	Team Member	GS	g.staiano11@studenti.unisa.it



## Sommario

<b>1. Introduzione</b>	<b>5</b>
<b>2. Relazione con altri documenti</b>	<b>5</b>
<b>3. Panoramica del sistema</b>	<b>5</b>
<b>4. Feature da testare</b>	<b>6</b>
<b>5. Pass/Fail criteria</b>	<b>6</b>
<b>6. Approccio</b>	<b>6</b>
7.1 Motivi di Sospensione	7
7.2 Ripresa dei Test	7
<b>8. Materiali di Testing</b>	<b>7</b>
8.1 Jest - Testing d'unità e Code Coverage	7
<b>9. Test Cases</b>	<b>9</b>
9.1 Gestione account	9
9.1.1 TC Registrazione	9
9.1.2 TC Login	9
9.1.3 TC Visualizzazione Account	9
9.2 Gestione Tutorial	10
9.2.1 TC Creazione Tutorial	10
9.2.2 TC Visualizzazione lista Tutorial	10
9.3 Gestione Feedback	11
9.3.1 TC Creazione Feedback	11
9.3.2 TC Visualizza Feedback	11
9.4 Gestione Quiz	12
9.4.1 TC Creazione Quiz	12
9.4.2 TC Svolgimento Quiz	12



## 1. Introduzione

Il presente documento definisce il piano di test per la piattaforma Tech4All, una piattaforma educativa online che offre tutorial e quiz per l'apprendimento delle basi dell'informatica. L'obiettivo è garantire la qualità e la funzionalità della piattaforma, assicurando un'esperienza utente fluida e efficace.

Questo documento presenterà un piano dettagliato delle attività di testing che verranno condotte per garantire la qualità della piattaforma. Saranno specificate le strategie di test, le funzionalità da verificare e gli strumenti utilizzati per minimizzare gli errori e ottimizzare le performance.

## 2. Relazione con altri documenti

Il presente documento, che costituisce una referenza completa per tutti i materiali prodotti fino a questo momento, è soggetto a future revisioni in seguito all'elaborazione di ulteriori documenti. I casi di test in esso contenuti sono strettamente correlati alle funzionalità descritte nel RAD (Requirements Analysis Document) e sono strutturati in linea con la suddivisione in sottosistemi presentata nell'SDD (System Design Document). Tale allineamento garantisce una copertura completa dei requisiti funzionali e non funzionali del sistema.

## 3. Panoramica del sistema

L'architettura del sistema si basa su un modello a tre livelli, favorendo una netta separazione tra interfaccia utente, logica di business e persistenza dati. Per lo sviluppo dell'interfaccia, abbiamo scelto React, framework JavaScript che ci permette di creare interfacce utente dinamiche e reattive in modo efficiente e organizzato. Le tecnologie HTML5 e CSS3 sono state utilizzate per strutturare e stilizzare le pagine web. Sul lato server, abbiamo optato per Node.js e TypeScript, un potente abbinamento che ci consente di sviluppare un back-end scalabile e facilmente manutenibile. Per la gestione del database, abbiamo selezionato MySQL, un sistema di gestione di database relazionale ampiamente utilizzato e noto per la sua flessibilità e affidabilità. MySQL ci offre una soluzione robusta per la persistenza dei dati, garantendo prestazioni elevate e facilità di gestione.

## 4. Feature da testare

Di seguito la lista delle feature di cui si effettuerà il testing per le varie gestioni:

- Gestione account:
  - Registrazione;
  - Login;
  - Visualizzazione account;
- Gestione Tutorial:
  - Creazione tutorial;
  - Visualizzazione lista Tutorial;
- Gestione feedback:
  - Creazione feedback;
  - Visualizza feedback;
- Gestione quiz:
  - Creazione quiz;
  - Svolgimento quiz;

I test si focalizzeranno sulle funzionalità ad alta priorità che richiedono un input diretto da parte dell'utente, escludendo quelle automatiche o con un impatto minore sul sistema.

## 5. Pass/Fail criteria

L'obiettivo del testing è verificare la conformità del sistema ai requisiti funzionali. Per ogni test case, viene definito un oracolo, ovvero il risultato atteso. Un test è considerato passato quando l'output effettivo devia dall'oracolo, indicando la presenza di un fault. Al contrario, un test fallito conferma il corretto funzionamento del sistema rispetto al requisito specifico. Per garantire la qualità del software, è necessario coprire tutti i requisiti funzionali ad alta priorità e effettuare test di regressione per prevenire la regressione dei difetti. Inoltre, è richiesto un livello minimo del 75% di branch coverage per assicurare un'adeguata copertura dei percorsi di esecuzione del codice.

## 6. Approccio

Per garantire la correttezza di ciascuna componente del software, eseguiremo dei test di unità. Utilizzando un approccio white-box, analizzeremo il codice sorgente di ogni unità per verificarne il corretto funzionamento in base ai requisiti specificati. Questo ci permetterà di costruire una solida base per i successivi livelli di testing.

## 7. Sospensione e Ripresa

### 7.1 Motivi di Sospensione

L'esecuzione dei test prosegue ininterrottamente fino a quando tutte le funzionalità previste non sono state validate. Tuttavia, potrebbero verificarsi situazioni che richiedono una temporanea interruzione dei test:

- Rilevamento di un difetto critico: Se viene individuato un problema che impedisce il corretto funzionamento del sistema, i test vengono sospesi per consentire la risoluzione immediata del difetto.
- Introduzione di nuove funzionalità: Quando vengono implementate nuove funzionalità, i test possono essere temporaneamente sospesi per riallinearli alle nuove specifiche e garantire una copertura completa.

### 7.2 Ripresa dei Test

I test riprendono non appena:

- Tutti i difetti sono stati risolti: Una volta risolti i problemi identificati durante la fase di sospensione, i test riprendono per verificare l'effettiva correzione dei difetti e l'impatto sulle altre funzionalità.
- Le nuove funzionalità sono state completamente implementate: Una volta che le nuove funzionalità sono state sviluppate e integrate nel sistema, vengono eseguiti nuovi test per verificarne il corretto funzionamento e l'interazione con le funzionalità esistenti.

## 8. Materiali di Testing

### 8.1 Jest - Testing d'unità e Code Coverage

Jest è stato utilizzato per il testing di unità e l'analisi della copertura del codice nel progetto, grazie alla sua capacità di verificare in modo rapido ed efficace le funzionalità implementate. Questi test hanno confermato che i comportamenti definiti nei requisiti sono stati correttamente implementati, senza riscontrare anomalie o errori.

Jest è stato sfruttato nelle sue funzionalità nei seguenti modi separatamente:

- Per il testing di unità è stato utilizzato tramite il terminale con il comando specifico `npm test` seguito dal nome del file della test suite specifica che si desiderava testare. Questo ha semplificato di molto il lavoro consentendo di verificare singolarmente il corretto svolgimento dei Test Case presenti al loro interno.
- Per il Code Coverage è stato possibile fin da subito seguire i progressi che le singole suite apportavano alla copertura totale del codice, grazie al comando da terminale `npm run testr`. Tale comando genera infatti un report di coverage su tutto il sistema visualizzato in



forma di percentuale, riportando tra l'altro anche le righe di codice non ancora testate, fornendo un supporto nella definizione e affinamento dei test proposti. Per facilitare l'esecuzione dei test relativi alle services, è stato creato uno script dedicato all'interno del file package.json, testserv, che consente di visualizzare in maniera unificata la coverage dei soli Services, permettendo di avere un focus maggiore sul progresso del testing di unità svolti.

Grazie a questi strumenti e al loro utilizzo mirato, è stato possibile garantire un'elevata affidabilità, confermando che soddisfa pienamente le specifiche richieste.



## 9. Test Cases

In questa sezione definiremo i casi di test, che saranno poi specificati nel dettaglio all'interno del documento "Test Case Specification".

Si fornirà la tabella con i nomi ed id dei test cases, senza particolari specifiche.

### 9.1 Gestione account

#### 9.1.1 TC Registrazione

Test Case Id	Nome
TC_2.2.1	Email già registrata
TC_2.2.2	Formato password non valido
TC_2.2.3	Nome non valido
TC_2.2.4	Cognome non valido
TC_2.2.5	Formato email non valido
TC_2.2.6	Registrazione completata

#### 9.1.2 TC Login

Test Case Id	Nome
TC_2.4.1	Email errata
TC_2.4.2	Email e password mancanti
TC_2.4.3	Formato email non valido
TC_2.4.4	Formato password non valido
TC_2.4.5	Lunghezza email non valida
TC_2.4.6	Lunghezza password non valida
TC_2.4.7	Lunghezza email non valida
TC_2.4.8	Autenticazione completata

#### 9.1.3 TC Visualizzazione Account

Test Case Id	Nome
TC_2.6.1	Visualizzazione corretta
TC_2.6.2	Utente non esiste
TC_2.6.3	Id Utente non fornito
TC_2.6.4	Visualizzazione utenti
TC_2.6.5	Errore durante restituzione



## 9.2 Gestione Tutorial

### 9.2.1 TC Creazione Tutorial

Test Case Id	Nome
TC_3.2.1	Lunghezza del titolo invalida
TC_3.2.2	Formato del titolo non corretto
TC_3.2.3	Formato della grafica non corretto
TC_3.2.4	Lunghezza del testo invalida
TC_3.2.5	Categoria non trovata
TC_3.2.6	Creazione tutorial completata

### 9.2.2 TC Visualizzazione lista Tutorial

Test Case Id	Nome
TC_3.4.1	Lista Tutorial vuota
TC_3.4.2	Errore durante la restituzione della lista dei Tutorial
TC_3.4.3	Lista Tutorial restituita correttamente

### 9.3 Gestione Feedback

#### 9.3.1 TC Creazione Feedback

Test Case Id	Nome
TC_4.2.1	Recensione vuota
TC_4.2.2	Valutazione assente
TC_4.2.3	Lunghezza recensione non valido I
TC_4.2.4	Lunghezza recensione non valido II
TC_4.2.5	Feedback esistente
TC_4.2.6	Creazione Feedback completata

#### 9.3.2 TC Visualizza Feedback

Test Case Id	Nome
TC_4.4.1	UserID mancante
TC_4.4.2	TutorialID mancante
TC_4.4.3	Feedback non trovato
TC_4.4.4	Feedback trovato

## 9.4 Gestione Quiz

### 9.4.1 TC Creazione Quiz

Test Case Id	Nome
TC_5.2.1	Lunghezza domanda non valida I
TC_5.2.2	Lunghezza domanda non valida II
TC_5.2.3	Lunghezza risposta non valida I
TC_5.2.4	Lunghezza risposta non valida II
TC_5.2.5	Domanda vuota
TC_5.2.6	Risposta vuota
TC_5.2.7	Creazione del quiz corretta

### 9.4.2 TC Svolgimento Quiz

Test Case Id	Nome
TC_5.4.1	Corretto Completamento
TC_5.4.2	ID Quiz mancante
TC_5.4.3	Id Quiz inesistente
TC_5.4.4	ID Utente mancante
TC_5.4.5	Errore esecuzione generico
TC_5.4.6	Tutorial Id valido
TC_5.4.7	Tutorial Id non valido
TC_5.4.8	Quiz inesistente per tutorial
TC_5.4.9	Eliminazione riuscita
TC_5.4.10	Errore esecuzione eliminazione
TC_5.4.11	Id Quiz non esistente