



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# **Il progetto del corso di Tecnologie Web - A.A. 2023/24**

**Angelo Di Iorio, Fabio Vitali,  
Andrea Schimmenti, Lorenzo Paolini**

Corso di laurea in Informatica per il  
Management

Alma Mater – Università di Bologna

# Premessa

- Gli studenti di Informatica hanno ricevuto una presentazione di progetto simile a questa
- I progetti sono compatibili tra loro ma si differenziano in rapporto ai CFU del corso nel piano di studi
- La regola si applica anche ai corsi mutuati: per un corso da 6CFU fate riferimento a questo progetto, per un corso da 9CFU al progetto per Informatica
- Gli studenti di Informatica per il Management e i corsi mutuati devono seguire queste specifiche



# Progetto in team

- Progetto di gruppo, formato da 2-3 persone
- Attenzione: meno di 2 significa troppo lavoro individuale. Più di 3 significa troppo poco. AL MASSIMO 3. NIENTE ECCEZIONI.
- Ogni team porta il progetto insieme (anche qui, non ci sono eccezioni!). Il team dichiara in anticipo la natura del contributo di ciascun membro oppure accetta che chiunque sia interrogato (e nel dettaglio) su tutto il progetto.
- Il sottoscritto NON è coinvolto nell'organizzazione dei team.



# Ruolo di queste specifiche

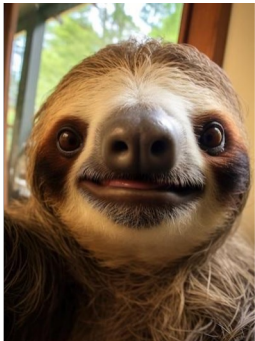
- Questo documento contiene le specifiche fondamentali del progetto di fine corso.
- Quanto scritto in nero, salvo esplicite eccezioni, deve essere considerato requisito OBBLIGATORIO per la consegna.
  - Le frasi scritte in arancione si riferiscono a funzionalità opzionali per migliorare la valutazione, e sono **non obbligatorie**.
  - Le frasi in verde corrispondono a vincoli **obbligatorie** introdotti solo per le esigenze del progetto universitario, non necessari o opportuni in un prodotto vero per il mercato esterno.
- Se una o più delle specifiche qui introdotte non funzionano, il progetto NON è considerato accettabile.
- ***Su Virtuale forniremo eventuali modifiche ai requisiti.***





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# SELFIE



# Selfie: fondamenti

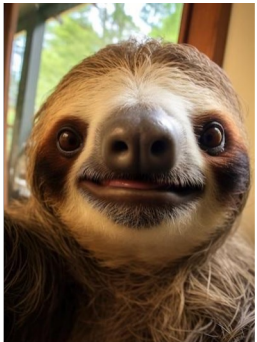
La vita di uno studente universitario è ricca e complessa, con eventi, scadenze, impegni che possono molto diversi tra loro:

- *individuali o di gruppo,*
  - *unici* (un aperitivo con gli amici questo venerdì alle 19:00) o *ripetuti* (il calcetto del martedì alle 20:00),
  - *semplici* (appello di TW il 12 giugno alle 11:00) o *complessi e strutturati* (dal 10 al 21 maggio studiare la teoria del progetto, dal 22 maggio al 5 giugno preparare il progetto, sera del 6 giugno demo con i compagni di gruppo, 7-11 debug del progetto, 7-11 studio dell'esame scritto, 12 esame scritto di TW), etc. Spesso anche sovrapposti.
- Impegni individuali o di gruppo
- Chiedere se un evento più grande , si deve dividere in più task(es.:laurea(dividere questo evento in piccoli task))

Attraverso SELFIE lo studente UniBo è in grado di programmare la propria vita, privata, sociale ed accademica, in maniera flessibile e completa.

Selfie è un'applicazione client + server usabile in maniera equivalente sia da cellulare che da PC, organizzata in moduli che forniscono supporto a tipi diversi di eventi, scadenze ed impegni della vita dello studente.





# Selfie : Architettura

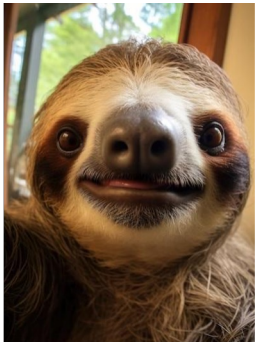
L'applicazione SELFIE prevede un'applicazione client-side in Javascript e/o Typescript collegata in maniera precisa con una parte server-side

Prevede quattro moduli (View) obbligatori:

- **Calendario**: permette di aggiungere, rimuovere, postare e modificare eventi semplici del solo utente, posizionati in un calendario e di durata nota (intervallo di ore, intere giornate, periodi più lunghi. Gli eventi si possono sovrapporre liberamente. Esistono visualizzazioni comode giornaliere, settimanali e mensili degli eventi inseriti.
- **Timer**: (la view Pomodoro) per organizzare il passo dello studio
- **Note**: per prendere appunti e note di studio
- **Time machine**: un sistema per navigare nel tempo e arrivare ad una data passata o futura come desiderato

A questi si aggiunge una Home e almeno DUE view opzionali, scelte e configurate dal gruppo



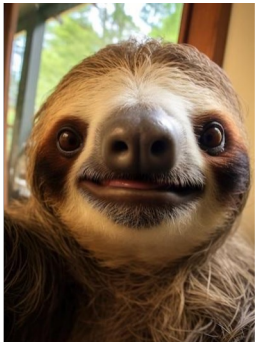


# Home page e Accesso Utente

- L'utente accede all'app tramite account basato su **nome utente e password**.
- Il record di un account contiene sicuramente nome utente, password e nome vero ed una quantità a piacere di informazioni personali (e.g. selfie personale, data del compleanno da aggiungere al calendario, ecc.).
- La Home page serve da Hub per navigare tra le view: Calendario, Pomodoro, Note, **<Altre View>**.
- Nell'Home vengono **mostrate preview** dei contenuti delle singole view: ad esempio gli eventi della settimana/giorno corrente, l'ultima nota creata, le attività imminente, report sull'ultimo pomodoro svolto, ecc.
- Gli utenti hanno la possibilità di personalizzare il tipo di preview per ogni view (i dettagli della personalizzazione sono a discrezione del gruppo)





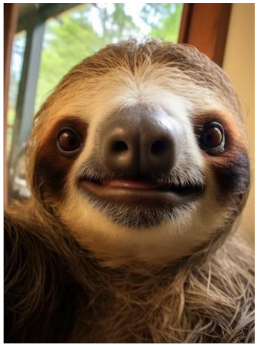


# Calendario - Eventi

Il calendario consente ad un utente di creare eventi: questi sono specifici appuntamenti o incontri che hanno una data, un'ora e una durata definite. Hanno un titolo.

- Possono essere **una tantum**, come un appuntamento medico, o ripetibili, come lezioni settimanali. Possono anche essere lunghi come tutta la giornata o più giorni.
- Gli **eventi ripetibili** hanno
  - una frequenza (es.: *tutti i giorni, tutti i martedì e giovedì, una volta la settimana, tutti i giorni 4 del mese, tutti i primi lunedì del mese, ecc.*)
  - un numero di ripetizioni (*ripeti indefinitamente, ripeti N volte, ripeti fino alla data XX/XX/XXXX*).
- Gli eventi possono anche includere un luogo fisico o virtuale dove si svolgono.
- Nel caso di luogo fisico SELFIE mostra informazioni di geolocalizzazione.



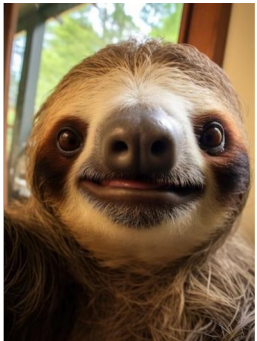


# Calendario - Attività

Specie di “to do list”  
che può essere associato a un intervallo di tempo

- Il calendario consente ad un utente di creare attività: queste sono azioni o compiti di durata prolungata e non esclusiva che un utente deve completare mentre fa anche altre cose.
- Non necessariamente devono svolgersi in un momento specifico, ma possono avere una scadenza. Ad esempio, "completare la relazione" ed inviarla entro una certa data.
- In genere non sono associate ad un intervallo di tempo preciso, ma "da adesso ed entro una certa data".
- Il loro completamento deve essere esplicitamente specificato o si trascinano nei giorni successivi (attività in ritardo)
- Le attività devono essere visualizzate nel calendario come scadenza e separatamente come lista.



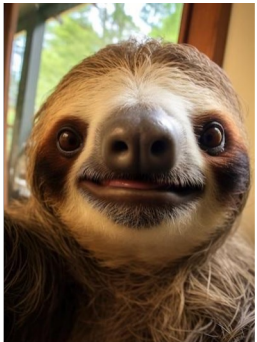


# Le Note

Chiedere al prof se le persone che hanno l'accesso possono scrivere oppure no nelle note (noi vorremmo fare in modo che l'autore possa scegliere chi ha l'autorizzazione a scrivere e chi può solo leggere )

- Una **nota** è un testo di lunghezza arbitraria (fate prove anche con testi molto lunghi) dotata di titolo, categorie (a scelta dell'utente), data di creazione, e data di ultima modifica.
- Le note devono essere gestite in una view separata dal calendario. **E' possibile duplicare note** e copiare ed incollare il contenuto delle note. E' possibile cancellare note.
- L'home delle note deve mostrare una preview delle note esistenti (i primi N caratteri, N almeno 200) e consentire di aggiungerne delle nuove. Possono essere categorizzate per ordine alfabetico (dal titolo), per data e per lunghezza del contenuto.
- **Le note possono essere scritte con un editor WYSIWYG o in markdown, a discrezione del gruppo (per markdown si veda ad esempio <https://www.npmjs.com/package/marked>)**
- **(se non si implementa questa funzionalità le note sono private). Le note sono dotate di autore e lista di accesso: aperte a tutti, aperte ad alcune specifiche persone, note di gruppo**





# Timer – pomodoro

La view Timer gestisce il metodo di studio *pomodoro* per studiare. Organizza il tempo dello studente in cicli studio-relax secondo una tecnica fissa di 30+5 minuti.

La view è composta da:

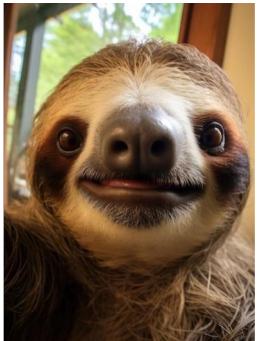
- Form per scegliere il tempo di studio e di pausa. Quello standard è 5 cicli da 30 minuti di studio e 5 minuti di pausa.
  - Si possono anche inserire un totale di ore/minuti disponibili e si ottengono una o più proposte di cicli di studio/pausa
- Inizio del tempo di studio/pausa successivo forzato da bottone;
- Tasto ricomincia ciclo;
- Tasto fine ciclo.
- Notifica per inizio ciclo, passaggio da una fase alla successiva, fine ciclo.

Si richiede un'animazione (**OBBLIGATORIAMENTE** fatta in CSS, non una gif) per lo studio e una per la pausa

E' possibile creare Eventi speciali sul calendario, associati alla view pomodoro.

Cliccando sull'evento si viene rimandati a questa view





# Calendario – Notifiche\*

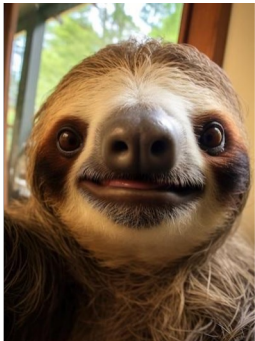
*\* View opzionale, testo della slide in nero solo per questioni di leggibilità*

SELFIE include un'area di notifica. I messaggi compaiono mentre sto usando l'app, ma anche mentre sto facendo qualcos'altro. La messaggistica deve essere calibrabile per testo, ripetizione, urgenza crescente, gestione dei ritardi e degli snooze.

Gli utenti possono indicare meccanismo di notifica dell'imminente appuntamento

- usando uno o più meccanismi (area di notifica, mail, Telegram, ecc.)
- con un certo anticipo (all'ora voluta, un minuto, cinque minuti, un'ora, due ore, un giorno, due giorni prima, ecc.)
- con una certa ripetizione (ripeti tre volte, ripeti ogni minuto, ripeti ogni ora, ripeti fino a che non rispondo, ecc.).





# Calendari avanzati\*

\* *View opzionali*, testo della slide in nero solo per questioni di leggibilità

## Calendari di gruppo

- Gli utenti possono includere altri utenti nei loro eventi
- Gli altri utenti possono accettare, rifiutare o rimandare l'accettazione dell'evento
- Gli utenti possono indicare certi intervalli di tempo (anche ripetuti) come non disponibili per eventi di gruppo
- I calendari di gruppo possono includere attività

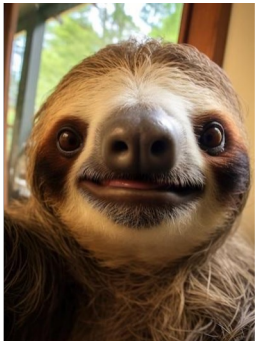
## Calendari integrati

- Gli eventi devono essere integrati con lo standard iCalendar, ed essere liberamente importabili/esportabili. (Google Calendar, Apple Calendar, ecc. anche via mail.

Alcuni esempi di librerie che aiutano a gestire i file iCalendar:

- <https://www.npmjs.com/package/ical>
- <https://www.npmjs.com/package/datebook>
- <https://www.npmjs.com/package/@pgswe/ics.js>





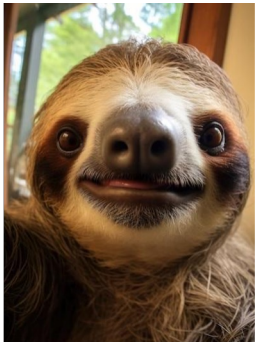
# La Time Machine

Per testare il sistema abbiamo bisogno di poter viaggiare nel tempo!

Questo significa che OGNI ANNOTAZIONE TEMPORALE DI QUALUNQUE TIPO sia del server che del client non dipende dalla data e dall'ora del sistema operativo, ma dalla data e dall'ora di un apposito servizio chiamato *Time Machine*.

- per default è allineato alla data e all'ora del sistema operativo ma è possibile modificare data e ora in avanti ed indietro in qualunque momento.
- Una apposita parte dell'interfaccia, separata dalle altre view ma sempre visibile, e con colori contrastanti rispetto al resto dell'interfaccia, permette all'utente di cambiare data ed ora della Time Machine,
- Immediatamente e senza reload le view cambiano e riflettono la nuova data.
- Le notifiche relative al giorno appena specificato si attivano, ma non quelle dei giorni precedenti.
- Un semplice pulsante rimette a posto la view alla data e ora del sistema operativo.





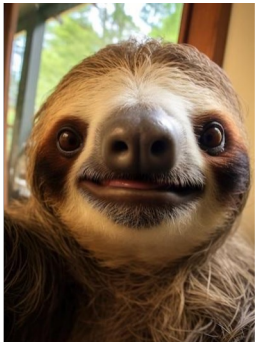
# Gestione progetti\*

*\* View opzionale, testo della slide in nero solo per questioni di leggibilità*

- Un progetto è una lista di attività in sequenza o parallele attribuite ad uno o più attori. Le attività sono raggruppate in fasi e sottofasi.
- Le attività possono partire quando è disponibile un input (anche vuoto) e producono un output (anche solo un booleano "attività completata").
- Se l'output dell'attività X è l'input dell'attività Y, le due attività sono sincronizzate e i problemi di X ricadono sullo svolgimento di Y sotto forma di traslazione (le due settimane sono ancora due settimane ma finiscono dopo) o contrazione (le due settimane diventano una settimana e mezzo per rispettare la data di conclusione originaria).
- Alcuni output sono importanti e sono chiamati *milestones*. Le attività che generano milestones hanno delle date di conclusione precise e non flessibili (quindi non possono spostarsi ma solo contrarsi).







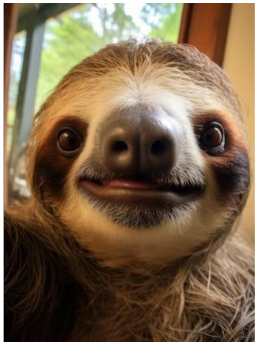
# Gestione progetti\*

\* *View opzionale, testo della slide in nero solo per questioni di leggibilità*

Ogni attività è associata ad un valore di status continuamente aggiornato e affidabile:

- *Non attivabile* (non è ancora disponibile l'input relativo)
- *Attivabile* (ma non attivata: l'input è presente ma l'attore non ha ancora dichiarato di averla iniziata)
- *Attiva* (l'attore ha dichiarato di averla iniziata)
- *Conclusa* (l'attore ha dichiarato di averla conclusa ed un output è disponibile)
- *Riattivata* (il capo-progetto ha rifiutato l'output e ha richiesto revisioni)
- *In ritardo* (la data di conclusione è passata ma l'output non è ancora disponibile)
- *Abbandonata* (la data di conclusione è passata da molto tempo, oppure l'attore ha dichiarato di non volersene più occupare)





# Gestione progetti\*

*\* View opzionale, testo della slide in nero solo per questioni di leggibilità*

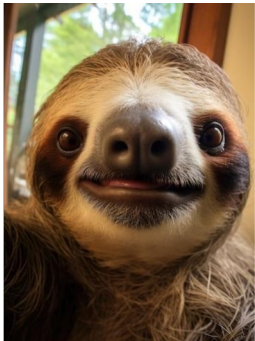
La visualizzazione è organizzata con le seguenti regole:

- Inizio e fine attività sono eventi che finiscono sul calendario con modalità di visualizzazione separate dagli eventi normali ed appropriate allo scopo.
- Due visualizzazioni speciali per i progetti sono sempre disponibili: una a lista (ordinata sia temporalmente sia per attore coinvolto) e una a GANTT.

Diversi attori lavorano al progetto con ruoli diversi:

- Chiunque può creare progetti (e diventa il capo-progetto) e coinvolgere altri utenti.
- Solo il capo-progetto può modificare il progetto, aggiungere, togliere o raggruppare attività e cambiare le sincronizzazioni tra le attività. Solo il capo-progetto in caso di ritardi può decidere se traslare o contrarre le attività sincronizzate a quella in ritardo.



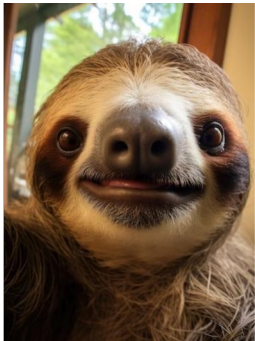


# Gestione progetti\*

Supponiamo oggi sia il 17 aprile 2024

| Anno                              |          |              |              |           | 2024   |    |    |    |        |    |    |    |        |    |    |    |
|-----------------------------------|----------|--------------|--------------|-----------|--------|----|----|----|--------|----|----|----|--------|----|----|----|
| Mese                              |          |              |              |           | aprile |    |    |    | maggio |    |    |    | giugno |    |    |    |
| Fase e attività                   |          |              |              |           | gg     | 03 | 10 | 17 | 24     | 01 | 08 | 15 | 22     | 29 | 05 | 12 |
| <b>Preparazione Esame scritto</b> |          |              |              |           |        |    |    |    |        |    |    |    |        |    |    |    |
| Studio teoria                     | io       | 01/05        | 07/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Studio esercizi HTML              | io       | 08/05        | 14/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Studio esercizi Javascript        | io       | 15/05        | 21/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Studio esercizi Framework         | io       | 22/05        | 28/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Ripasso globale                   | io       | 29/05        | 11/06        | 13        |        |    |    |    |        |    |    |    |        |    |    |    |
| <i>Esame scritto</i>              |          | <i>12/06</i> |              |           |        |    |    |    |        |    |    |    |        |    |    |    |
| <b>Preparazione Progetto</b>      |          |              |              |           |        |    |    |    |        |    |    |    |        |    |    |    |
| <i>Ricezione specifiche</i>       |          | <i>10/04</i> |              |           |        |    |    |    |        |    |    |    |        |    |    |    |
| Progetto base                     | tutti    | 10/04        | 30/04        | 20        |        |    |    |    |        |    |    |    |        |    |    |    |
| Test progetto base                | tutti    | 30/04        |              |           |        |    |    |    |        |    |    |    |        |    |    |    |
| <b>Realizzazione modulo I</b>     | io       | <b>02/05</b> | <b>21/05</b> | <b>19</b> |        |    |    |    |        |    |    |    |        |    |    |    |
| <b>Test modulo I</b>              | io       | <b>22/05</b> | <b>28/05</b> | <b>6</b>  |        |    |    |    |        |    |    |    |        |    |    |    |
| Test integrazione base + mod I    | tutti    | 29/05        | 04/06        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Realizzazione modulo II           | Andrea   | 02/05        | 21/05        | 19        |        |    |    |    |        |    |    |    |        |    |    |    |
| Test modulo II                    | Andrea   | 22/05        | 28/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Test integrazione base + mod II   | tutti    | 29/05        | 04/06        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Realizzazione modulo III          | Beatrice | 02/05        | 21/05        | 19        |        |    |    |    |        |    |    |    |        |    |    |    |
| Test modulo III                   | Beatrice | 22/05        | 28/05        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Test integrazione base + mod III  | tutti    | 29/05        | 04/06        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| Integrazione completa e deploy    | tutti    | 05/06        | 11/06        | 6         |        |    |    |    |        |    |    |    |        |    |    |    |
| <i>Consegna progetto</i>          |          | <i>12/06</i> |              |           |        |    |    |    |        |    |    |    |        |    |    |    |





# Gestione degli eventi

*\* parte opzionale, testo della slide in nero solo per questioni di leggibilità*

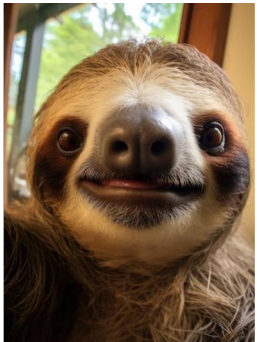
Questa parte è facoltativa per tutti e di grande difficoltà (creativa, non realizzativa). Verrà fatta molta attenzione alla semplicità e flessibilità del meccanismo di inserimento, spostamento e modifica degli eventi e delle attività.

- **Efficacia:** attenzione alla facilità a commettere errori, alla precisione, alla possibilità di lavorare su molti eventi insieme
- **Efficienza:** tempo e numero di azioni atomiche necessarie per richiedere la modifica ed effettuarla
- **Soddisfazione:** sofisticazione grafica e chiarezza di visualizzazione della modifica e degli eventi/attività modificate

## Hints

- Le ore del giorno non sono tutte uguali: pochi impegni tra mezzanotte e le 8, e dalle 8 alle 18 sono per lo più impegni di lavoro (spesso cadenzati e regolari), dopo le 18 sono per lo più impegni del tempo libero. Stessa cosa per giorni lavorativi e weekend.
- Le notifiche delle attività in ritardo debbono prevedere un meccanismo crescente di urgenza e disturbo, in modo che diventi sempre maggiore la pressione sull'utente.





# View obbligatorie ed estensioni

- Ogni gruppo è tenuto ad implementare:
  - View obbligatorie: Home, Calendario, Timer, Note
  - Time Machine
- Oltre ad ALMENO DUE tra le seguenti estensioni:
  - Sistema di notifica
  - Calendari di gruppo
  - Calendari esterni
  - Sistema di gestione di progetti complessi
  - Qualunque altra estensione non descritta in queste slide ma decisa e configurata dal gruppo, anche di complessità ridotta:
    - Chat real-time
    - Social network
    - Podcast e musica
    - Tornei
    - Sondaggi
    - ...
    - *[spazio alla fantasia! Molto apprezzata!]*



# Vincoli Mobile-first e PC-first

- Le parti **calendario, note e pomodoro** sono **mobile** first, sono realizzate con il framework Javascript/Typescript e CSS preferito, e sono pensate per essere usate velocemente e facilmente da tutti. Su PC è comunque possibile compiere in maniera adeguata le funzionalità previste.
- La Time Machine è sempre visualizzata su PC e immediatamente accessibile (senza navigazione) su mobile.
- La parte gestione Progetto (**se inclusa**) è PC-first, e sfrutta **completamente e appropriatamente uno schermo grande**





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Regole e consegna

# Esame e progetto

- L'esame consiste di due prove:
  - **Prova di laboratorio:** domande di teoria ed esercizi sul programma svolto, in laboratorio - voto massimo di 27
  - **Discussione** di un **progetto** di gruppo - voto da 0 a 5
- Il progetto è **opzionale**
- Il voto complessivo si ottiene dalla somma dei voti:  
**Voto laboratorio (max 27) + [ voto progetto (max 5) ]**
- I voti sono normalizzati se il progetto è stato già consegnato o la prova di laboratorio già superata nei precedenti a.a.
- Per dettagli consultare le slide di introduzione al corso





# Lavoro di team e contributo individuale

- Tutti i membri dei team sono tenuti a lavorare e lavorare insieme.
- E' meglio essere parte attiva di un progetto mediocre che passiva di un progetto meraviglioso.
- Non saranno tollerati i portatori di pizze
- Mi riservo all'esame di scoprire il **contributo individuale di ciascuno**, indipendentemente dalla bontà del progetto consegnato
- Tutti i componenti del gruppo devono dare un contributo significativo. Non è sufficiente, ad esempio, lavorare solo a regole CSS o solo alle funzionalità di login
- Si parte da una valutazione complessiva del progetto ma il voto di ognuno può variare in modo significativo in base al contributo personale (fino a consegnare di nuovo se il contributo è insufficiente)



# A proposito di contributo individuale

- Esistono molte librerie che offrono le funzionalità di questo progetto, molte open-source
- Ed ovviamente esiste ChatGPT...
- Si possono riusare librerie esistenti (consigliato)
- Si può usare ChatGPT ma l'uso deve essere limitato a problemi ripetitivi
- L'uso deve essere documentato esplicitamente. Lo stesso vale per le librerie
- I membri del gruppo debbono comunque essere **consapevoli** e **padroni** del loro codice
  - capiscono cosa fa ogni singola riga del codice
  - sono in grado di mettervi mano e modificarlo a piacimento senza AI
- All'esame scritto ChatGPT non c'è, e verranno richieste competenze che richiedono di sapere scrivere codice corretto e in velocità

# Discussione

- La discussione consiste nel mostrare il funzionamento dell'applicazione e il codice sorgente
- Esame in presenza o da remoto. Tutti i membri del gruppo devono essere presenti
- L'applicazione dovrà essere installata su un vostro computer e avere alcuni dati pre-caricati
- E' richiesto avere a disposizione un ambiente per aggiornare il codice durante la discussione
- E' ammesso usare soluzioni che richiedono compilazione ma anche in questo vi verrà chiesto di modificare l'applicazione durante la discussione
- La discussione è parte integrante della valutazione del progetto. Non viene valutato solo il codice.



# Consegna

- Il progetto si consegna su Virtuale 5 giorni prima della discussione
- Saranno pubblicati spazi appositi per la consegna ad ogni appello
- Cosa consegnare:
  - Sorgenti (o un link per scaricarli)
  - README con i componenti del gruppo
  - Istruzioni per installare e usare l'applicazione (possono essere molto brevi, anche una sola riga!)
- Contestualmente alla consegna, iscriversi su AlmaEsami per la corrispondente discussione
  - Iscrivere una sola persona per gruppo



# Vincoli sul progetto: back-end e front-end

- Il focus è su **interazione client-side** ed **integrazione di componenti**
- Il backend è minimale **e non è obbligatorio l'uso di un database.**
- Il backend essere scritto in un qualunque linguaggio di programmazione ma noi vediamo solo Node/ExpressJS
- **La gestione del database è meno importante per gli scopi di questo progetto, rispetto ad interazione client-side, API e mashup**
- Il codice client-side può essere scritto in plain Javascript o con JQuery o altri framework (component-based)
- L'uso di un framework non è obbligatorio ma valutato positivamente



# Vincoli sul progetto: persistenza

- E' ammesso salvare i dati direttamente sul filesystem (es. in JSON) o usare database SQL/NoSQL (ma che non vedremo)
- In questo secondo caso si può:
  - usare un servizio online con dati pre-caricati e consegnare un'applicazione già configurata per collegarsi al servizio, senza richiedere l'installazione di un database in locale
  - fornire istruzioni e dati per popolare un database locale
- Includere nella consegna dati ragionevoli per mostrare il funzionamento dell'applicazione: utenti, eventi, note, messaggi



# Altre regole

- Il progetto si consegna e viene valutato una sola volta ma se insufficiente bisogna riconsegnarlo
- Le specifiche del progetto sono valide per questo a.a. (quindi fino a Febbraio 2025). Nel prossimo a.a. il progetto cambierà.
- Il voto resta valido per l'intero anno accademico e per gli anni successivi (almeno finché ci sarò io)
- I voti dei progetti discussi in passato restano validi, dopo la normalizzazione, per l'intero a.a. 2023/24 (almeno)



# Consiglio finale

- Le prove di laboratorio e il progetto sono separati per non imporre vincoli MA è consigliato portarli avanti in parallelo
- Iniziate a lavorare anche al progetto durante il corso Lavorare al progetto aiuta a prepararsi alla prova di laboratorio e viceversa
- L'ultima data utile per consegnare questo progetto è febbraio 2025: non arrivate all'ultimo appello!
- Vale anche per la prova di laboratorio 😊





# Domande?

