

به نام خدا



## درس برنامه سازی پیشرفته

فاز سوم پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۹۹-۰۰

---

استاد:

دکتر محمد امین فضلی

مهلت ارسال:

مسئول پروژه:

امیر مهدی نامجو

مسئول فاز دوم:

طراحان فاز دوم:

مسئولین تنظیم مستند:

پارسا محمدیان و سروش جهانزاد

# فهرست

## نکات قابل توجه

۲

## مقدمه

۳

## بخش های اصلی

۳

|   |                              |
|---|------------------------------|
| ۳ | احراز هویت                   |
| ۴ | اتاق گفت و گو (Chat Room)    |
| ۴ | تابلوی امتیازات (Scoreboard) |
| ۴ | بازی دو نفره                 |
| ۵ | لابی                         |
| ۵ | فروشگاه                      |

## بخش های امتیازی

۶

|    |  |
|----|--|
| ۶  | اتاق گفت و گو (Chat Room)                                |
| ۶  | دستاوردها (Achievements)                                 |
| ۷  | بخش تلویزیون (TV)  |
| ۸  | امتیازدهی به بازی ها                                     |
| ۸  | بازی دو نفره   |
| ۹  | محدودیت زمانی هر نوبت                                    |
| ۹  | امکان بازگشت به بازی بعد از قطع شدن اتصال کلاینت با سرور |
| ۹  | مزایده در فروشگاه  |
| ۱۰ | نمایش بازیکنان آنلاین در اسکوربرد                        |
| ۱۰ | امکان ارسال دعوتنامه برای بازی دو نفره                   |
| ۱۱ | نیاز نداشتن به رفرش                                      |
| ۱۱ | استفاده از دیتابیس                                       |



## نکات قابل توجه

- پس از اتمام این فاز، در گیت خود یک تگ با ورژن "v3.0.0" بزنید. در روز تحویل حضوری این tag بررسی خواهد شد و کدهای پس از آن نمره‌ای نخواهد گرفت. برای اطلاعات بیشتر در مورد شیوه ورژن‌گذاری، می‌توانید به [این لینک](#) مراجعه کنید. البته برای این پروژه صرفاً رعایت کردن همان ورژن گفته شده کافیست، اما خوب است که با منطق ورژن‌بندی هم آشنا بشوید.
- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره‌ی ایشان برای آن فاز پروژه "صفر" لحاظ می‌گردد. مشارکت، با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه بررسی می‌شود.
- در هر فاز می‌توانید سه روز تاخیر به ازای کسر نمره داشته باشید که به ازای هر روز آن، ۱۰ درصد از نمره آن فاز را از دست خواهید داد. در مجموع سه‌فاز پروژه، سه روز تاخیر نیز بخشیده خواهد شد.
- به ازای هر ساعتی که پروژه را زودتر تحویل دهید، ۱۵ دقیقه به مهلت تاخیر بدون کسر نمره شما اضافه خواهد شد. این مقدار حداکثر یک روز خواهد بود که در صورت ارسال ۴ روز زودتر از ددلاین به شما تعلق خواهد گرفت. **بنابراین ددلاین‌های پروژه تحت هیچ شرایطی تمدید نخواهد شد.** توصیه می‌شود با برنامه‌ریزی مناسب به ددلاین‌های درس پایبند باشید.
- در صورت کشف تقلب از هریک از تیم‌ها، برای بار اول منفی نمره آن فاز برای آن تیم ثبت می‌شود و برای بار دوم، نمره منفی کل پروژه برای تیم لحاظ خواهد شد که معادل مردود شدن در درس است.



## مقدمه

همانطور که میدانید، در دو فاز قبلی بدنه‌ی اصلی مورد نیاز برای اجرای بازی روی یک کامپیوتر را پیاده‌سازی کردیم؛ یعنی بخش‌های مربوط به منطق و گرافیک بازی که برای کامل بودن بازی کافی هستند. پس در این بخش می‌خواهیم چه کنیم؟ قرار است قابلیت‌هایی را به بازیمان اضافه کنیم تا بشود بر بستر اینترنت هم بازی را اجرا و با بقیه بازیکنان از راه دور بازی کرد. همچنین این فاز بخش‌های امتیازی متنوع و زیادی دارد که برای جبران نمرات از دست رفته در دو فاز قبلی می‌توانید از آن‌ها استفاده کنید. در ادامه بخش‌های اصلی و امتیازی را به صورت جداگانه می‌توانید ببینید.

## بخش‌های اصلی

### احراز هویت

فرض کنید در دنیای واقعی می‌خواهید وارد گروهی شوید که برای اعضایش قابلیت‌های ویژه‌ای در نظر گرفته شده است. شما ثبت‌نام می‌کنید و مسئولین گروه مشخصات شما را یادداشت می‌کنند؛ سپس به شما کارت عضویت داده می‌شود. شما با این کارت احراز هویت می‌شوید و می‌توانید از شرایط خاص گروه بهره‌مند شوید.

احراز هویت در برنامه‌نویسی نیز خاصیت مشابهی دارد. شما پس از ثبت‌نام و احراز هویت در سایت مورد نظر، دارای حساب کاربری می‌شوید و می‌توانید از ویژگی‌های خاص آن سایت استفاده کنید.

حال سوالی که مطرح می‌شود این است که احراز هویت به چه شکل صورت می‌گیرد؟ در ابتدا کاربر باید ثبت‌نام کند. به صورت عادی از کاربران نام کاربری و رمز عبور خواسته می‌شود (اطلاعات فرد). پس از تایید ثبت‌نام، کاربر دارای حساب کاربری شده و می‌تواند با نام کاربری و رمز عبور وارد حساب خود شود (کارت عضویت).

در برنامه‌های رایانه‌ای، سرور برای پردازش هر درخواست (request) از سمت کلاینت، نیاز دارد که کاربر درخواست‌دهنده را بشناسد.

عموماً از دو روش زیر سرور متوجه هویت کاربر حاضر می‌شود.

- بر پایه‌ی نشست (session):

در این روش سرور برای هر کاربری که وارد می‌شود (login می‌کند) یک نشست می‌سازد و درخواست‌هایی که از آن دستگاه به سرور می‌آیند را در نشست مورد نظر پردازش می‌کند و از این راه متوجه هویت کاربر درخواست‌دهنده می‌شود.

- بر پایه‌ی توکن (token):

در این روش پس از ورود کاربر، سرور یک توکن (token) به کلاینت ارسال می‌کند.



از این پس کلاینت در کنار هر درخواست این توکن را هم به سمت سرور می‌فرستد تا سرور متوجه هویت کاربر درخواست‌کننده شود.

ما به شما پیشنهاد می‌کنیم که از احراز هویت برپایه‌ی توکن (token-based authorization) استفاده کنید.

از لینک‌های زیر می‌توانید برای مطالعه بیشتر استفاده کنید:

- [تفاوت‌های این دو روش احراز هویت](#)

- [بررسی این دو روش در برابر هم](#)

- [احراز هویت برپایه‌ی توکن](#)

## اتاق گفت‌وگو (Chat Room)

همانطور که در فاز یک اشاره شده بود، در این فاز قصد داریم ویژگی اتاق گفت‌وگو را به بازی خود اضافه کنیم.

هدف اصلی ما اضافه کردن اتاقی عمومی به برنامه است تا تمام افرادی که در برنامه ثبت‌نام کرده‌اند بتوانند در آن صحبت و گفت‌وگو کنند. در بخش اصلی تنها باید اتاق گفت‌وگوی عمومی (چت همگانی یا Global Chat) را پیاده‌سازی کنید.

### نکته

اضافه کردن ویژگی «گفت‌وگوی خصوصی» و یا «گفت‌وگو در حین بازی» جزو خواسته‌های اصلی این قسمت نیست و نمره امتیازی نیز نخواهد داشت؛ اما برای جذاب‌تر شدن هرچه بیشتر برنامه می‌توانید در صورت داشتن وقت و حوصله کافی (: این ویژگی‌ها را نیز به برنامه خود اضافه کنید.

## تابلوی امتیازات (Scoreboard)

پیاده‌سازی تابلوی امتیازات باید به گونه‌ای باشد که اطلاعات آن در سمت سرور نگهداری شود و هر بار کلاینت برای دیدن نتایج به سرور درخواست بدهد.

## بازی دو نفره

در این نوع از بازی، بازیکنان باید بتوانند به صورت‌های مختلف (شامل ۱ دور و ۳ دور) با یکدیگر مسابقه دهند. به بیان دیگر، در این بخش همان بازی دو نفره که در فاز ۱



پیاده‌سازی کردید باید در دو کلاینت مختلف به صورت هم‌زمان اجرا شوند و هر بازیکن در یک کلاینت جداگانه به بازی وارد شود و بازی را انجام دهد. در این حالت هر کلاینت اتفاقات مدنظر بازیکن را در نوبت خودش به سمت سرور می‌فرستد و در صورت مجاز بودن خواسته‌ها، آن اتفاقات عملی می‌شوند و سرور پاسخ مناسب را در صورت نیاز به کلاینت‌ها می‌فرستد.

همچنین واضح است که بازیکنان در نوبت حریف نیز می‌توانند به صفحه‌ی بازی دسترسی داشته باشد و به طور هم‌زمان کارهایی که حریف انجام می‌دهد را ببیند؛ البته قوانین اصلی باید رعایت شوند و بخش‌هایی از صفحه‌ی بازی که فقط در دسترس خود بازیکن است باید از حریف پنهان باشند. مثلاً کارت‌هایی که در دست حریف قرار دارند، نباید قابل مشاهده باشند.

## لابی

برای اینکه بازیکنان بتوانند همدیگر را برای مسابقه دادن پیدا کنند به یک لابی نیاز داریم. در لابی بازیکنان می‌توانند با یکدیگر تعامل داشته باشند (می‌توانید قسمت چت عمومی را نیز به لابی اضافه کنید). ویژگی دیگر لابی این است که هر بازیکن می‌تواند برای بازی دوفره درخواست بدهد و نوع بازی مد نظرش را در آن درخواست مشخص کند. پس از آن، اگر شخص دیگری هم درخواستی مشابه او داشت سرور برای آنها یک بازی می‌سازد و آنها را به هم متصل می‌کند.

توجه کنید تا زمانی که درخواست مشابهی وجود نداشت، بازیکن در حالت انتظار باقی می‌ماند، مگر این که به اختیار خودش از حالت انتظار خارج شود. همچنین اگر چندین نفر به طور هم‌زمان درخواست‌های مشابه داشتند، سرور آنها را به صورت تصادفی به یکدیگر متصل می‌کند.

## فروشگاه

در این فاز فروشگاه بازی باید به سرور منتقل شود و تمامی تغییراتی که روی اقلام فروشگاه رخ می‌دهد در سرور ذخیره می‌شوند. همچنین تمامی کارت‌هایی که در فروشگاه موجودی مشخصی خواهند داشت و اگر بازیکنی کارتی را خرید، موجودی آن کارت در فروشگاه یک واحد کم می‌شود. در صورتی که موجودی کارتی تمام شد، هیچ بازیکنی دیگر نمی‌تواند آن کارت را بخرد. از طرفی بازیکن‌ها می‌توانند کارت‌هایی که دارند را بفروشند و در این صورت موجودی آن کارت‌ها در فروشگاه به تعداد فروخته‌شده بیشتر می‌شود.

علاوه بر این، باید برای فروشگاه یک پنل ادمین ساده ایجاد کنید تا بتوان آن را مدیریت کرد. از طریق این پنل می‌شود خرید و فروش کارت‌های به خصوص را ممنوع کرد. به عنوان مثال اگر از کارت X به مقدار ۵ تا در فروشگاه موجود باشد ولی از پنل مدیریت خریدش



ممنوع شده باشد، کارت همچنان در فروشگاه نمایش داده می‌شود اما یک علامت ممنوعه روی آن زده می‌شود تا نشانگر این وضعیت باشد و امکان خرید آن وجود نخواهد. کارهای ساده‌ی دیگر مانند افزودن یا کم کردن موجودی یک کارت نیز از این پنل در دسترس هستند.

## بخش‌های امتیازی

### اتاق گفت‌وگو (Chat Room)

علاوه بر اتاق گفت‌وگوی عمومی، می‌توانید بخش‌های زیر را هم به صورت امتیازی پیاده‌سازی کنید.

- قابلیت حذف پیام
- قابلیت ویرایش (edit) پیام
- قابلیت پاسخ دادن (reply) پیام
- قابلیت سنجاق کردن (pin) پیام
- نمایش آواتار ارسال کننده پیام
- نمایش اطلاعات کاربری فرد با کلیک بر روی آواتار او
- نمایش تعداد افراد آنلاین (آنلاین به این معنا که در حال حاضر در برنامه لاگین کرده و حضور دارند، نه این‌که لزوماً در صفحه مربوط به اتاق گفت‌وگو باشند!).

### دستاوردها (Achievements)

بازیکنان در طول اجرای بازی مهارت‌های مختلفی بدست خواهند آورد و تجربیات و کنش‌های مختلفی را پشت سر خواهند گذاشت. به پاس قدردانی از زحمات آن‌ها، نشان‌هایی برای آن‌ها در نظر بگیرید که اگر موفق به انجام کنش مورد نظر شدند، به آن‌ها تعلق بگیرد و در پروفایل کاربری آن‌ها درج شود. این کنش‌ها را در ادامه مطرح می‌کنیم.

هم‌چنین علاوه بر در نظر گرفتن نشان می‌توانید وابسته به سختی این کنش‌ها، مقداری پاداش برای هر عمل در نظر بگیرید. این پاداش می‌تواند به صورت اهدای یک کارت به بازیکن و یا اهدای مقداری پول و یا امتیاز به او باشد.

### تعریف کنش‌ها

در ادامه تعدادی کنش مطرح می‌شود که مقداردی به پارامترهای آن‌ها پیشنهادی است اما به ذوق و سلیقه خودتان می‌توانید هر عدد معقولی بگذارید.



- دنباله‌ای متوالی از پیروزی‌ها به تعداد مناسب (مثلاً ۵، ۱۰، ۱۵ و ...)
- پیروزی در تعداد مناسبی (مثلاً ۵) مسابقه فقط با استفاده از کارت‌های هیولا
- دنباله‌ای متوالی از پیروزی‌ها با محدودیت کاهش سلامتی مثلاً سه برد متوالی با سلامتی بیش از ۳/۴ مقدار اولیه
- پیروزی در تعداد محدودی نوبت
- دنباله‌ای متوالی از شکست‌ها به تعداد مناسب
- اولین - دهمین - صدمین و ... پیام اتاق گفت‌وگو
- احضار کارت به صورت آیینی (Ritual Summon)
- ساخت اولین زنجیره

#### نکته

دقت کنید که برخی از موارد بالا را می‌توان به صورت لحظه‌ای (real time) پیاده‌سازی کرد؛ به این معنا که بلافاصله بعد از برآورده شدن شرط مربوطه، دستاورد به کاربر تعلق بگیرد و پیام مناسب نمایش داده شود؛ پیاده‌سازی این موارد به صورت لحظه‌ای نمره امتیازی خواهد داشت. امتیازدهی به کنش‌ها نیز متناسب با سختی پیاده‌سازی آن‌ها خواهد بود.

### بخش تلویزیون (TV)

صفحه‌ای را فرض کنید که دارای سه قسمت (tab) پخش زنده، بازپخش و پخش بازی‌های بازیکنان برتر باشد. از آنجا که منطق کلی هر بخش با بخش دیگر یکسان است و همه شامل نمایش یک بازی می‌شوند، می‌توان همه را در یک صفحه با نام تلویزیون (TV) گنجاند. در ادامه به شرح هر سه بخش می‌پردازیم.

#### • پخش زنده‌ی بازی‌ها

هدف این قسمت به اشتراک‌گذاری برخط بازی‌هاست و در واقع مانند استریم بازی‌ها عمل می‌کند. هر کاربری که در حال بازی کردن باشد می‌تواند پخش زنده‌ی بازی خود را به اشتراک بگذارد. روش پیشنهادی ما برای این بخش ثبت گزارش (Log) است. این روش به صورت زیر قابل اجراست.

ابتدا حالت اولیه بازی (شامل کارت‌های هر بازیکن و اسپل‌ها و ...) ذخیره می‌شود. سپس هر حرکتی که رخ بدهد ذخیره می‌شود. به عبارت دیگر پس از هر اتفاق، جزئیات





آن در قالب مشخصی ذخیره می‌شود. سپس چون حالت اولیه‌ی بازی را می‌دانیم برای هر کاربری که قصد مشاهده‌ی پخش زنده را داشته باشد یک نمونه از بازی می‌سازیم و تمامی حرکات را براساس این گزارش‌ها اجرا می‌کنیم تا به حالت فعلی بازی برسیم. پس از آن نیز به طور مداوم با استفاده از ادامه‌ی گزارش‌ها پخش زنده ادامه پیدا می‌کند.

### • مشاهده‌ی دوباره‌ی بازی

کاربر می‌تواند در این بخش هر بازی خود را دوباره مشاهده و بررسی کند. به کمک روش ثبت Log می‌توان هر بازی را ذخیره کرد و در هر حرکت، به حرکت‌های قبلی و بعدی در بازی دست یافت. همچنین این حرکت‌ها می‌توانند به صورت خودکار و مانند یک فیلم طبیعی، به صورت خودکار و پشت سر هم پخش شوند.

### • مشاهده بازی‌های ضبط شده‌ی بازیکنان برتر

بازی بازیکنان برتر ذخیره می‌شود و بعدها نیز می‌توان این بازی‌ها را مشاهده کرد. این بخش هم به کمک ثبت Log می‌تواند پیاده‌سازی شود. این بازی‌ها در کنار اسکوربرد قابل دسترسی هستند. نگهداری چند بازی اخیر بازیکنان برتر کافیهست و نیازی نیست بازی‌ها تا ابد نگهداری شوند.

## امتیازدهی به بازی‌ها

برای این بخش باید قابلیت در نظر بگیرید تا بینندگان بتوانند به بازی‌های برتر که در بخش تلویزیون نشان داده می‌شوند امتیاز بدهند. برای این بخش نکات زیر را در نظر داشته باشید.

- داشتن سیستم امتیازدهی ساده
- به منظور قابل اطمینان بودن این امتیاز، باید به نوعی صلاحیت فرد امتیاز دهنده احراز شود؛ مثلاً حتماً باید ۷۵ درصد از نوبت‌های بازی را تماشا کرده باشد (یا به هر روش دیگری که صلاح می‌دانید!)

## بازی دو نفره

### ساختن هوشمند بازی (Matchmaking):

می‌توانید در منطق متصل کردن بازیکنان در سرور، منطقی قرار دهید که بازیکنانی با فاصله‌ی امتیازی قابل توجه به یکدیگر متصل نشوند یا با زمان انتظار بیشتری متصل شوند. با این حال باید برای زمان تحویل این بخش نیز سناریویی طراحی کنید. به عبارتی تحویل



گیرنده‌ی پروژه باید متوجه ۳ موضوع شود، در غیر این صورت امتیاز این بخش را به صورت کامل دریافت نمی‌کنید:

- سرور بازیکنان با فاصله‌ی امتیازی را با تاخیر به هم متصل می‌کند.
- سرور بازیکنانی که امتیاز نزدیک به هم دارند را بدون وقفه به هم متصل می‌کند.
- اگر بازیکنان A و B با فاصله‌ی امتیازی قابل توجه در حال انتظار باشند و در این بین بازیکن C که امتیاز نزدیک به A دارد درخواست بازی داد، سرور A را به C متصل کند و B در حال انتظار بماند.

### محدودیت زمانی هر نوبت

هر نوبت (turn) محدودیت زمانی خاصی دارد و اگر بازیکنی که نوبت اوست حرکت خود را انجام ندهد و زمان نوبت به پایان برسد بازی را می‌بازد و بازیکن رقیب برنده می‌شود.

### امکان بازگشت به بازی بعد از قطع شدن اتصال کلاینت با سرور

اگر در میانه‌ی یک turn ارتباط بازیکنی که نوبت اوست با سرور قطع شود تا زمان دوباره وصل شدن ارتباط بازیکن به سرور زمان turn ثابت نگه داشته می‌شود و بعد از اتصال دوباره زمان شروع به حرکت می‌کند.

### مزایده در فروشگاه

یکی دیگر از ویژگی‌هایی که باید در این فاز به بازی اضافه کنید مزایده است. مزایده به این صورت است که هر بازیکن می‌تواند از کارت‌هایی که در اختیار دارد یک کارت را با تعیین قیمت اولیه‌ی آن برای فروش بگذارد. بازیکنان دیگر می‌توانند بیشترین هزینه‌ای را که تا به حال برای آن کارت پیشنهاد داده شده است (یا قیمت اولیه) را افزایش دهند در نهایت با اتمام زمانی که توسط سرور از قبل تعیین شده بازیکنی که بیشترین پیشنهاد را برای کارت داشته برنده‌ی مزایده می‌شود. در مورد نحوه‌ی کسر اعتبار از بازیکن برنده در ادامه توضیحاتی آورده‌ایم. برای همه‌ی مزایده‌ها مدت زمان مشخصی در نظر بگیرید همچنین هر بار که یک بازیکن قیمت پیشنهادی را افزایش می‌دهد زمان باقی‌مانده به مقدار مشخصی ریست شود. انتخاب این حداقل زمان و همچنین مدت زمان مزایده‌ها به سلیقه‌ی خودتان خواهد بود.

برای پیاده‌سازی این بخش علاوه بر اضافه کردن محلی برای شروع مزایده توسط هر بازیکن باید بخشی به بازی اضافه کنید که بازیکنان بتوانند مزایده‌های فعال را ببینند و وارد آن‌ها شود.



## بخش اضافه کردن مزایده

در این بخش بازیکن به کارت‌های در اختیارش دسترسی دارد و می‌تواند یکی از آنها را انتخاب کند، سپس تنها با انتخاب یک قیمت اولیه مزایده را شروع می‌کند.

## بخش مشاهده مزایده‌های فعال

باید در منوهای هر بازیکن صفحه‌ای باشد که لیست تمامی مزایده‌های فعال همراه با آخرین قیمت پیشنهاد شده، زمان باقی مانده و عنوان کارت قابل مشاهده باشد، بازیکنان در این صفحه باید بتوانند پیشنهاد کارت‌ها را افزایش دهند. توجه کنید که با افزایش قیمت پیشنهادی کارت‌ها از اعتبار بازیکن به اندازه‌ی قیمتی که پیشنهاد داده کم می‌شود و اعتبار بازیکن قبلی که پیشنهاد بالا را داده بود به او برگردانده می‌شود. این صفحه باید یک دکمه برای رفرش کردن قیمت‌ها و زمان‌های باقی مانده داشته باشد. (در صورتی که بخش امتیازی را انجام نمی‌دهید)

## نمایش بازیکنان آنلاین در اسکوربرد

وضعیت آنلاین بودن بازیکنان باید در scoreboard مشخص شود. برای این کار باید یک لیست از سوکت‌هایی که بازیکنان با آن به سرور متصل هستند نگه‌داری کنید (احتمالاً در بخش‌های دیگر پروژه نیز به این نیاز پیدا کرده‌اید/خواهید کرد) و هر بار که شخصی درخواست مشاهده‌ی scoreboard را داشت سرور باید بازیکنانی که سوکت آنها باز هست را آنلاین نشان دهد. همچنین در scoreboard یک دکمه‌ی رفرش قرار دهید که با استفاده از آن اطلاعات scoreboard از جمله وضعیت آنلاین بودن بازیکنان بروزرسانی شود.

## امکان ارسال دعوتنامه برای بازی دو نفره

بازیکنان باید بتوانند برای یکدیگر دعوتنامه بفرستند با این توضیحات که یکی از ویژگی‌هایی که هر بازیکن می‌تواند در هنگام مشخص کردن نوع بازی تعیین کند، اینست که می‌خواهد با چه کسی بازی کند. سپس مانند قبل منتظر پاسخ سرور می‌ماند. اگر بازیکنی که به او درخواست داده می‌شود این درخواست را بپذیرد سرور آن دو را به هم متصل می‌کند و اگر پیشنهاد را رد کند بازیکن درخواست دهنده از حالت انتظار خارج می‌شود. همچنین مانند قبل، بازیکن باید خودش نیز توانایی خارج شدن از حالت انتظار را داشته باشد در این حالت با توجه به سلیقه‌ی خودتان ممکن است که درخواست داده شده را حذف کنید یا هر امکان دیگری قرار دهید که اگر بازیکنی که به او درخواست داده می‌شود، درخواست را قبول کند متوجه منقضی شدن دعوتنامه شود.



## نیاز نداشتن به رفرش

در بخش مشاهده‌ی مزایده‌های فعال و scoreboard اگر امکانی را فراهم کنید که بدون نیاز به دکمه‌ی رفرش کردن صفحه، اطلاعات صفحه بروزرسانی شود امتیاز این بخش را دریافت می‌کنید و همچنین ملزم به پیاده‌سازی دکمه‌ی رفرش نخواهید بود. به عبارتی:

- در بخش مزایده‌های فعال: اگر بازیکن A در صفحه‌ی مزایده‌ها باشد و بازیکن B آیمبدون اینکه بازیکن A کاری انجام دهد باید مزایده‌ی اضافه شده را ببیند. همچنین اگر بازیکن C قیمت پیشنهادی برای مزایده‌ای را افزایش داد بازیکن A باید متوجه شود.

- در بخش scoreboard: اگر بازیکن A در صفحه‌ی scoreboard باشد و بازیکن B به بازی وارد شود بازیکن A باید متوجه این تغییر شود و همچنین اگر امتیاز بازیکن C افزایش یابد.

## استفاده از دیتابیس

پایگاه داده (database) مجموعه‌ای از داده‌ها است که به صورت منظم با ساختار خاصی درون کامپیوتر نگهداری شده است. پایگاه‌های داده معمولاً توسط سامانه‌های مدیریت پایگاه داده (DBMS) (ها) کنترل می‌شوند. شما می‌توانید پایگاه داده خود را درون یک فایل به صورت ساده نگهداری کنید اما مهندسان بسیاری روی DBMS ها کار کرده‌اند و با توجه به شرایط کامپیوترهای موجود DBMS هایی را توسعه داده‌اند که عملیات های معمول مانند جستجو، دریافت داده‌هایی با شرایط خاص و تغییر در پایگاه داده را با سرعت بالا و راحتی انجام دهند. در ادامه با برخی از انواع DBMS ها و نحوه استفاده از آن ها آشنا می‌شویم.

### • RDBMS

این خانواده از DBMS ها بسیار وسیع و در عین حال پر کاربرد هستند. RDBMS ها از زبان درخواست ساخت یافته (SQL) استفاده می‌کنند. در این دیتابیس ها داده‌ها در جدول‌هایی با ویژگی‌های مشخص ذخیره می‌شوند. بین این جداول روابطی (Relation) برقرار می‌شود و ما می‌توانیم عملیاتی مانند انتخاب، (Select) درج، (Insert) به روزرسانی (Update) و حذف (Delete) به همراه سایر ابزارهای مورد نیاز برای مدیریت پایگاه داده‌ها را در کنار هم داشته باشیم. RDBMS های معروف فراوانی وجود دارند و شما بسته به نیازتان از آنها استفاده می‌کنید. از جمله معروف‌ترین آن‌ها می‌توان به MySQL، PostgreSQL، SQLite، و ... اشاره کرد.

[در این لینک می‌توانید استفاده SQLite در جاوا را مشاهده کنید.](#)



## • NOSQL

با پیشرفت تکنولوژی و در نتیجه نیازهای بشر، دیگر RDBMS ها پاسخگوی همه‌ی نیازها نبودند. در نتیجه DBMS های NOSQL ساخته شد. این‌ها از زبان SQL پشتیبانی نمی‌کنند و در عوض قابلیت‌های دیگری مانند نگهداری پایگاه داده در چند سرور مجزا، جستجو در گراف و ... را دارند. Elastic - MongoDB - Cassandra - Redis Search از دیتابیس‌های شناخته شده و پرکاربرد NOSQL هستند.

[در این لینک میتوانید استفاده MongoDB در جاوا را مشاهده کنید.](#)

در این فاز شما محدودیتی در استفاده از دیتابیس‌ها ندارید و میتوانید از هر دیتابیس که می‌خواهید استفاده کنید.