

প্রোগ্রামিংয়ের বলদ টু বস

(১৩তম চ্যাপ্টার)

যাদের ভেরিয়েবল, array, if-else, ফাংশন, ক্লাস, অবজেক্ট, ডিকশনারি, এইসব সম্পর্কে ধারণা আছে তারা এইটা পড়ে বুঝতে পারবে।

১৩

বিপদে পড়লে আরামের শহরটা, অবজ্ঞার শহর হতে দুই সেকেন্ডের বেশি সময় লাগে না। এখানে আড্ডার সঙ্গী মিলে, কিন্তু বিপদের বন্ধু মিলে না। এ শহরে ১০০০ টাকা ধার পাওয়ার চেয়ে এক রাত থাকার জায়গা পাওয়া অনেক বেশি কঠিন। এ শহর টিভি, সোফা, ডাইনিং টেবিলের জন্য আলাদা আলাদা রুম রাখতে পারে। কিন্তু পরিচিত মানুষের থাকার জন্য ১০ ইঞ্চি জায়গা ছাড় দিতে পারে না। এখানে রিকশা, টেম্পো, বাস সবই আছে কিন্তু সময়মতো গন্তব্যে পৌঁছানোর নিশ্চয়তা নেই। জ্ঞান আহরণ করার ভার্শিটি আছে, কিন্তু জ্ঞান অ্যাপ্লাই করার চাকরি নেই। সিস্টেমে দুর্নীতি আছে, জোর-জবরদস্তি আছে। তবে এর মধ্যে যে জীবনের মোড় ঘুরিয়ে দেওয়ার চ্যালেঞ্জ গ্রহণ করতে পারে, সে ঠিকই জীবনের মোড় ঘুরিয়ে দিতে পারে।

অ্যাপ বানিয়ে করবে বড়াই

এসি মার্কেটে ঢুকে আস্তে আস্তে হাঁটতে লাগল রাশেদ। সঙ্গে আছে অস্তু। একটা দোকানের সামনে এসে রাশেদ বলতে শুরু করল, শোন, এত দিন যা যা শিখছস, সেগুলো শিখে নাকে তেল দিয়ে ঘুমানো যাবে না; বরং সিরিয়াস হয়ে সেগুলো কাজে লাগাতে হবে। কোথাও আটকে গেলে অন্য কারও হেল্প নিতে হবে।

তুই হচ্ছস বলদ আর তোর বন্ধু হচ্ছে যাঁড়। এখন তুই আর তোর বন্ধু যাঁড়ের কাজ হবে এই সামনের দোকানের জন্য একটা সফটওয়্যার বা অ্যাপ বানানো। খুব কঠিন বা বিশাল সফটওয়্যার না। বরং তুই যে জিনিসগুলো শিখছস, সেগুলো দিয়ে সিম্পল একটা সফটওয়্যার বানাবি। তবে সফটওয়্যারের মধ্যে কী কী ফিচার থাকবে, সেগুলো এই দোকানের ম্যানেজারের সাথে আলাপ করে জেনে নে।

১. দোকানের একটা নাম থাকবে। ওপরে সাইনবোর্ডে যে নাম লেখা আছে, সেই নাম।

২. কাস্টমার এসে কোনো একটা জিনিস আছে কি না জানতে চাইলে, সেটা আছে কি না, বলে দিতে হবে।

৩. কোনো জিনিসের দাম জিঙ্কস করলে, সেটার দাম বলে দিতে হবে।

৪. সারা দিনে মোট কত টাকা বিক্রি হইছে, সেটার একটা হিসাব রাখতে হবে। যাতে দোকানের ক্যাশে কত টাকা আছে, এটার মালিক জানতে চাইলে তাকে বলে দেওয়া যায়।

৫. কাস্টমার এক বা একাধিক জিনিস কিনতে চাইলে, দুইটা সিচুয়েশন হতে পারে। এক, কাস্টমার যতগুলো চাইছে ততগুলো স্টকে নাই। তখন কাস্টমারকে বলতে হবে, সরি, আপনি যতগুলো চাচ্ছেন ততগুলো আমাদের দোকানে নাই। দুই, কাস্টমার যতগুলো চাইছে তার সমান বা বেশি থাকলে তিনটা কাজ করতে হবে।

(ক) কাস্টমার মোট কত টাকা দিবে, সেটা হিসাব করতে হবে।

(খ) কাস্টমার যত টাকা দিছে, সেটা আজকের দিনে যত টাকা বিক্রি হইছে তার সাথে যোগ করতে হবে।

(গ) যে কয়টা বিক্রি হইছে সে কয়টা স্টক থেকে কমাতে হবে।

৬. দোকানে নতুন বা পুরান আইটেম ওঠানোর সিস্টেম থাকতে হবে।

(ক) যে আইটেম আগে থেকেই আছে, সেটা আরও কয়েকটা ওঠানো হলে, স্টকে আগে যে কয়টা ছিল তার সাথে এখন যে কয়টা ওঠানো হচ্ছে, সে কয়টা যোগ করে দিলেই হবে।

(খ) আর যে আইটেম আগে কখনোই দোকানে ছিল না, সেটা যোগ করতে গেলে প্রথমেই সেটা স্টকে যোগ করতে হবে। সেটার দাম লিখে রাখতে হবে। যাতে কাস্টমার দাম জানতে চাইলে সেটার দাম বলে দেওয়া যায়। আবার কেউ এসে এই নতুন জিনিস আছে কি না, জানতে চাইলে, আছে বলে উত্তর দিতে হবে।

ধাপাধপ কাজ শুরু করার আগে ওপরের ছয়টা ফিচার ও ফিচারের সঙ্গে লাগানো লেজ দুবার করে পড়। কোনটা বুঝতে অসুবিধা হলে কাজ শুরু করার আগে দোকানের ম্যানেজারের কাছ থেকে বিস্তারিত জেনে নে।

কোনো সফটওয়্যার বানাতে গেলে এমনভাবে বানানো হয় যাতে সেটা একাধিক মানুষ ব্যবহার করতে পারে। যেমন ধর, তুই দোকানের জন্য সফটওয়্যার বানাচ্ছস। এখন এটা এমনভাবে বানাবি যাতে একাধিক দোকানের কাছে বিক্রি করা যায়।

মাটির কড়াই, চামে করে নামের বড়াই

প্রথম ফিচারটা কিছুই না। একটা নাম দিতে চাইছে। নাম দেওয়া তোর কাছে কোনো ব্যাপারই না। জাস্ট একটা ক্লাস ডিক্লেয়ার করবি। কারণ তুই এই সফটওয়্যার একাধিক দোকানের কাছে বিক্রি করতে চাচ্ছস। আর ক্লাস ব্যবহার করলে, যখন যে দোকানের কাছে বিক্রি করবি, তখন সে দোকানের নাম দিয়ে তাদের জন্য একটা দোকান অবজেক্ট বানিয়ে ফেলবি। তাতেই তোর কাজ হয়ে যাবে। যে ক্লাস লিখবি সেই ক্লাসে name নামে একটা প্রপার্টি রাখবি। আর তুই যেহেতু দোকানের জন্য সফটওয়্যার বানাতে চাচ্ছস তাই এই ক্লাসের নাম দিবি স্টোর (Store)। তার ভিতরে constructor ফাংশনে একটা ইনপুট প্যারামিটার নিবি। যেটার নাম হবে name। তারপর name ইনপুট প্যারামিটার দিয়ে name প্রপার্টির মান সেট করে ফেলবি।

```
class Store {  
    constructor(name) {  
        this.name = name;  
    }  
}
```

কী আছে কী নাই, খবর জেনে যাই

সেকেন্ড ফিচারটা হচ্ছে, দোকানে কোনো আইটেম আছে কি না, বলে দিতে হবে। এইটার জন্য তোকে পাঁচটা ছোট ছোট কাজ করতে হবে।

১. কী কী আইটেম আছে সেটা Store ক্লাসের প্রপার্টিতে রাখতে হবে। যেহেতু দোকান চালু করার আগে কোনো আইটেম থাকবে না, সেহেতু constructor ফাংশনের ভিতরে items নামে একটা প্রপার্টি যোগ করে দিলেই হবে। যার মান হবে খালি একটা array। খালি array ডিক্লেয়ার করার সিস্টেম আগে থেকেই জানস। জাস্ট দুইটা থার্ড ব্র্যাকেট দিবি। তবে এই array আপাতত খালি হলেও পরে যখন নতুন আইটেম দোকানে ওঠানো হবে, তখন তাদের নাম এই array তে push করা হবে।

২. যেহেতু কোনো একটা আইটেম আছে কি না জানতে চাচ্ছে। তাই Store ক্লাসের ভিতরে আরেকটা মেথড ডিক্লেয়ার করতে হবে। যেটার নাম দিবি, আইটেম আছে কি না (isItemAvailable)। আর যে জিনিসটা আছে কি না জানতে চাচ্ছে, সেই জিনিসের নাম ইনপুট প্যারামিটার হিসেবে নিবি।

৩. ওই যে items নামে একটা array ডিক্লেয়ার করছিলি। সেটার মধ্যে যে জিনিসটা আছে কি না জানতে চাচ্ছে, সেটা খুঁজতে হবে। array-এর মধ্যে কোনো একটা জিনিস খোঁজার সিস্টেম তুই আগে থেকেই জানস। প্রথমেই array-এর নাম লিখতে হবে। এইখানে তোর array হচ্ছে ক্লাসের items প্রপার্টি। তাই this.items লিখবি। তারপর ডট চিহ্ন (.) দিয়ে indexOf লিখে প্রথম ব্র্যাকেটের ভিতরে জিনিসটার নাম লিখে দিবি।

৪. এই যে indexOf বের করলি সেটা একটা ভেরিয়েবলে রাখতে হবে। সেই ভেরিয়েবলের নাম দিবি itemIndex অর্থাৎ আইটেমটার ইনডেক্স বা পজিশন কত। আর তুই আগে থেকেই জানস indexOf এর মান -1 হলে বোঝা যাবে জিনিসটা array-এর মধ্যে নাই। আর যদি -1 ছাড়া অন্য কোনো মান হয়, তাহলে বোঝা যাবে জিনিসটা array-এর মধ্যে আছে।

৫. যেহেতু itemIndex ভেরিয়েবলের মান -1 কি না চেক করতে হবে, সেহেতু তোকে if-else ব্যবহার করতে হবে। সে জন্য if লিখে তারপর ব্র্যাকেটের ভিতরে itemIndex == -1 শর্ত দিবি। এই শর্ত সত্য হলে জিনিসটা নাই। তাই if-এর শর্তের পরের সেকেন্ড ব্র্যাকেটের ভিতর থেকে false রিটার্ন করে দিবি। যাতে যে এই মেথডকে কল করবে সে false রিটার্ন পেয়ে বুঝতে পারে, জিনিসটা নাই। আর যদি itemIndex এর মান -1 না হয়, তখন if-এর পরে সেকেন্ড ব্র্যাকেটের

ভিতরে না গিয়ে else-এর পরে সেকেন্ড ব্র্যাকেটের ভিতরে যাবে। তখন তুই বুঝতে পারবি জিনিসটা আছে। তাই else-এর ভিতর থেকে true রিটার্ন করে দিবি। তাহলে যে isItemAvailable মেথডকে যে কল করবে সে বুঝতে পারবে জিনিসটা আছে।

পুরা ক্লাসটা দেখতে নিচের মতো। আর isItemAvailable মেথডটা ভালো করে দেখ।

```
class Store {  
    constructor(name) {  
        this.name = name;  
        this.items = [ ];  
    }  
    isItemAvailable(name){  
        var itemIndex = this.items.indexOf(name);  
        if(itemIndex == -1){  
            return false;  
        }  
        else {  
            return true;  
        }  
    }  
}
```

তুই একটা ফিচার ডেভেলপ করছস। সেটার জন্য একটা মেথড লিখছস। এখন এই মেথড ঠিকমতো কাজ করে কি না, চেক করতে হবে। তবে চেক করার জন্য ছোট করে আরেকটা সিম্পল মেথড যোগ করে নে। যেটার নাম দিবি addItem এবং ইনপুট হিসেবে একটা নাম (name) নিবি। তারপর সেই নামটা Store ক্লাসের items প্রপার্টিতে push করে দিবি। তা ছাড়া addItem মেথডটা পরে ৬ নম্বর ফিচারের জন্য ঠিকমতো করে লেখা হবে। এখন চেক করার জন্য ছোট ও সহজ করে লিখ।

তারপর Store ক্লাস ব্যবহার করে habluStore নামে একটা অবজেক্ট বানাবি। অবজেক্ট বানানোর সময় ইনপুট হিসেবে Bolod-Shaar Fashion House দিবি। তারপর habluStore-এর addItem মেথডকে কল করে "shirt" ইনপুট ব্যবহার করবি। যাতে items নামক array-তে শার্ট যোগ হয়ে যায়। একইভাবে "pant" ইনপুট হিসেবে ব্যবহার করে addItem মেথডকে আরেকবার কল করবি। তাহলে দোকানে শার্ট ও প্যান্ট ওঠানো হয়ে যাবে। পুরা কোডটা দেখতে নিচের মতো হবে।

```
class Store {
    constructor(name) {
        this.name = name;
        this.items = [ ];
    }
    isItemAvailable(name){
        var itemIndex = this.items.indexOf(name);
        if(itemIndex == -1){
            return false;
        }
        else {
            return true;
        }
    }
    addItem(name){
        this.items.push(name);
    }
}

var habluStore = new Store("Bolod-Shaar Fashion House");
habluStore.addItem("shirt");
habluStore.addItem("pant");
```

এখন তোর কাজ হবে isItemAvailable ঠিকমতো কাজ করতেছে কি না, সেটা চেক করা। সেটা করার জন্য "shirt"-কে ইনপুট প্যারামিটার হিসেবে ব্যবহার করে habluStore-এর isItemAvailable মেথডকে কল কর। রিটার্ন হিসেবে

true পাওয়ার কথা। কারণ তুই একটু আগে shirt যোগ করছিলি। তারপর এমন একটা জিনিস চেক কর যেটা তোর দোকানে নাই। যেমন : তুই জুতা আছে কি না, সেটা চেক করার জন্য "juta" ইনপুট প্যারামিটার হিসেবে ব্যবহার করে isItemAvailable মেথডকে কল কর। আউটপুট হিসেবে false পাওয়ার কথা।

```
habluStore.isItemAvailable("shirt");
```

```
habluStore.isItemAvailable("juta");
```

এই মুহূর্তে www.habluderadda.com/console-এ যা। সেখানে ওপরের Store ক্লাস, সেটার ভিতরের দুইটা মেথড লিখে, তোর পছন্দের নাম দিয়ে একটা দোকান বানাবি। তারপর সেটাতে আইটেম যোগ করবি। আইটেম যোগ করা হলে চেক করে দেখবি isItemAvailable মেথডটা ঠিকমতো কাজ করে কি না।

প্রফেশনাল প্রোগ্রামিং করার সময়, এইভাবেই একটা একটা করে ফিচার ডেভেলপ করে। তারপর সেটা চেক করে দেখে। চেক করার পর ঠিকমতো কাজ করলে পরের ফিচারের কাজ শুরু করে। এখন তুইও পরের ফিচারের কাজ শুরু করে দে।

ব্র্যান্ডের নাম, বাড়ায় শুধু দাম

থার্ড ফিচারটা হচ্ছে কোন একটা জিনিসের দাম জানতে চাইলে সেটার দাম বলে দিতে হবে। এই কাজটা করতে হলে তোকে ছোট ছোট পাঁচটা স্টেপ ফলো করতে হবে। এই পাঁচটা স্টেপ পড়ার সময় একটু পরে যে Store ক্লাসের কোড দেওয়া আছে সেটার সাথে মিলিয়ে নিবি। তাহলে পড়তে এবং বুঝতে সুবিধা হবে।

১. যেহেতু নাম বললে দাম বলে দিতে হবে, সেহেতু নামের সাথে দামের একটা সম্পর্ক আছে। আর কোনো একটা জিনিসের সাথে আরেকটা জিনিসের সম্পর্ক থাকলে key-value হিসেবে চিন্তা করা হয়। এইখানে key হচ্ছে নাম আর value হচ্ছে দাম। এইটুকু বুঝতে পারলে Store ক্লাসের constructor-এর ভিতরে একটা প্রপার্টি যোগ কর। যেটার নাম দিবি prices আর তার মান হিসেবে ডিকশনারি ডিক্লেয়ার করার জন্য দুইটা সেকেন্ড ব্র্যাকেট দিবি।

২. দাম কত সেটা বলে দেওয়া যেহেতু একটা কাজ, সেহেতু তোকে একটা মেথড ডিক্লেয়ার করতে হবে। যেটার নাম দিবি getPrice; আর যে আইটেমটা আছে কি না জানতে চাচ্ছে, সেটার নাম (name) ইনপুট প্যারামিটার হিসেবে নিতে হবে। তাই getPrice মেথডে name নামে একটা ইনপুট প্যারামিটার রাখবি।

৩. দাম বলে দেওয়ার আগে একখান কথা আছে। সেটা হচ্ছে, যেই জিনিস দোকানে নাই, সেটার দাম তো বলতে পারবি না। যেমন : জামাকাপড়ের দোকানে এসে কেউ পাউরুটির দাম জানতে চাইলে সেটা তো বলতে পারবি না। তাই দাম চেক করার আগে একটু আগে লেখা `isItemAvailable` মেথডকে কল করবি। কোনো একটা ক্লাসের একটা মেথডের ভিতরে থেকে সেই ক্লাসের অন্য কোনো মেথডকে কল করা খুবই সোজা। জাস্ট `this` লিখবি তারপর ডট চিহ্ন দিয়ে যে মেথডকে কল করতে চাস তার নাম লিখে কল করে দিবি। কল করার পর সেটা যে মান দিবে, সেই মান `isAvailable` নামে একটা ভেরিয়েবলে রাখবি।

৪. `if-else` দিয়ে চেক করবি `isAvailable`-এর মান `true` কি না। যদি `true` হয়, তাহলে সেকেন্ড ব্র্যাকেটের ভিতরে `itemPrice` নামে একটা ভেরিয়েবল ডিক্লেয়ার করবি। যেটার মান হবে এই ক্লাসের `prices` নামক ডিকশনারি থেকে বের করা এই জিনিসের দাম। আর কোনো একটা ডিকশনারি থেকে `key` ব্যবহার করে মান বের করার সিস্টেম তুই আগে থেকেই জানস। জাস্ট ডিকশনারির নাম লিখবি। এইটা যেহেতু একটা ক্লাসের মধ্যে তাই `this.prices` লিখবি তারপর থার্ড ব্র্যাকেটের ভিতরে নাম বসাই দিবি। তারপর `itemPrice`-কে রিটার্ন করে দিবি।

৫. `isAvailable`-এর মান `true` না হলে, জিনিসটা স্টোরে থাকবে না। তখন `if`-এর ভিতরে না গিয়ে `else`-এর ভিতরে যাবে। তাই সেখানে আউটপুট হিসেবে লিখে দে, সরি, আমাদের কাছে এই জিনিসটা নাই (Sorry. We do not have the item.)

এক্সট্রা, এইবার তুই দাম দেখানোর জন্য একটা মেথড লিখছস। তাই একটু আগে আইটেম যোগ করার জন্য `addItem` নামে যে মেথডটা আগে লিখছিলি, সেটা একটু মডিফাই করে নিতে হবে। আগে একটা ইনপুট নিছিলি। এখন দুইটা ইনপুট নিতে হবে। একটা হবে আইটেমের নাম (`name`) আরেকটা হচ্ছে দাম (`price`)। নাম আগের মতো `items` নামক `array`-তে `push` করে দিবি। আর `price`টা তোর `prices` ডিকশনারিতে রাখবি। ডিকশনারিতে রাখার সিস্টেম হবে `this.prices` লিখে থার্ড ব্র্যাকেটের ভিতরে নাম (`name`) লিখবি। তারপর সমান চিহ্ন দিয়ে দাম বসিয়ে দিবি। তাহলেই কাজ হয়ে যাবে।

পুরা ক্লাসটা দেখতে নিচের মতো হবে। ক্লাস ডিক্লেয়ার করার পর আগের মতো একটা দোকান অবজেক্ট বানিয়ে `addItem` মেথড কল করে `shirt` যোগ করবি। সাথে শার্টের দাম দিয়ে দিবি ৩০০ টাকা। একইভাবে `pant` যোগ করবি এবং প্যান্টের দাম দিবি ৫০০ টাকা।

```

class Store {
    constructor(name) {
        this.name = name;
        this.items = [ ];
        this.prices = {};
    }
    isItemAvailable(name){
        var itemIndex = this.items.indexOf(name);
        if(itemIndex == -1){
            return false;
        }
        else {
            return true;
        }
    }
    getPrice(name){
        var isAvailable = this.isItemAvailable(name);
        if(isAvailable == true ){
            var itemPrice = this.prices[name];
            return itemPrice;
        }
        else {
            console.log("Sorry. We do not have the item.");
        }
    }
    addItem(name, price){
        this.items.push(name);
        this.prices[name] = price;
    }
}

var habluStore = new Store("Bolod-Shaar Fashion House");
habluStore.addItem("shirt", 300);
habluStore.addItem("pant", 500);

```

এখন তোর কাজ হবে এই getPrice মেথডটা ঠিকমতো কাজ করছে কি না, সেটা চেক করা। সেটা করার জন্য প্রথমেই pant-কে ইনপুট প্যারামিটার হিসেবে ব্যবহার করে getPrice মেথডকে কল করবি। আউটপুট হিসেবে 500 পাওয়ার কথা। তারপর পাউন্ডটির দাম জানতে চাওয়ার জন্য bread-কে ইনপুট

প্যারামিটার হিসেবে ব্যবহার করে আবারও getPrice মেথডকে কল কর। এইবার Sorry. We do not have the item. আউটপুট হিসেবে পাওয়ার কথা।

```
habluStore.getPrice("pant");
```

```
habluStore.getPrice("bread");
```

আগের মতো এইবারও www.habluderadda.com/console-এ চলে যা। সেখানে পুরা ক্লাসটা লিখে getPrice মেথড ঠিকমতো কাজ করতেছে কি না, চেক করে দেখ।

সবাই মিলে করি কাজ, কাজ শেষে দিব নাচ

মোট ছয়টা ফিচার ছিল। যেখানে তুই তিনটা ডেভেলপ করছস। বাকি তিনটা ডেভেলপ করছে তোর পার্টনার মিস্টার ষাঁড়। এই যে আরেকজন কোনো কিছু ডেভেলপ করছে, তার কোড দেখে বুঝতে পারাটাও একটা কোয়ালিটি। আর পুঁচকা প্রোগ্রামাররা প্রথম প্রথম অন্যের কোড দেখে কিছুই বুঝতে পারে না। তুই যদি পাঁচবার দেখে অর্ধেক জিনিসও বুঝতে পারস, আমি গ্যারান্টি দিলাম তুই বলদ থেকে বস হয়ে যেতে পারবি। অল্প কয়দিনে ষাঁড়ের চেয়েও সরস হয়ে যাবি।

চতুর্থ ফিচার হচ্ছে মালিক জানতে চাইলে, ক্যাশে কত টাকা আছে বলে দিতে হবে। এই ফিচারটা খুবই মামুলি একটা ফিচার। এইটার জন্য তোর বন্ধু মিস্টার ষাঁড় প্রথমই মোট বিক্রি (totalSales) নামে একটা প্রপার্টি যোগ করছে। যেটার প্রাথমিক মান শূন্য। পরে যখন কোনো কিছু বিক্রি হবে, তখন totalSales-এর মান বাড়ানো হবে। তারপর টোটাল কত বিক্রি হইছে, সেটা বলে দেওয়ার জন্য getTotalSales নামে একটা মেথড লিখে দিছে। এই getTotalSales-এর কাজ কিছুই না। জাস্ট totalSales প্রপার্টির মান রিটার্ন করা। তোর মন চাইলে getTotalSales মেথডটা একটু পরের Store ক্লাসে গিয়ে দেখে আসতে পারস।

কাস্টমার, ত্যানাতুনাকে টাকায় কনভার্ট করার ট্রান্সফরমার

পঞ্চম ফিচারে হালকা ভেজাল আছে। এই আড্ডার শুরুতে গিয়ে, ৫ নম্বর ফিচারটা আরেকবার ভালো করে দেখে আয়। তারপর একটু পরের Store ক্লাসে sellItem মেথডটা দেখে দেখে বোঝার চেষ্টা কর। দেখে দেখে না বুঝলে মন খারাপ করার কিছুই নাই। একটু পরে আমি বলে দিব। তবে তুই আগে নিজে নিজে বোঝার চেষ্টা কর।

দেখে আসছস তো? এইবার আমি বুঝাই দিচ্ছি। শোন, সে মেথডটার নাম দিচ্ছে sellItem; sell মানে বিক্রি করা আর item মানে জিনিস। অর্থাৎ কোনো একটা জিনিস বিক্রি হলে প্রোগ্রামের ভিতর যে যে কাজ করা লাগবে, সেগুলো sellItem মেথডের ভিতরে করছে। সেগুলো যে ধাপে ধাপে ডেভেলপ করছে, সে ধাপগুলোকে সাতটা ভাগে ভাগ করা যায়। সাতটা ধাপ শুনে ভয় পাইস না আবার। এগুলো খুবই সিম্পল সিম্পল সাতটা ধাপ। তাই ইজি মুডে আগাইতে থাক।

১. এই ফিচারের প্রথমেই আছে কাস্টমার যতগুলো চাইছে, ততগুলো স্টকে না থাকলে সরি বলে দিতে হবে। তার মানে স্টকে কয়টা থাকবে সেটার একটা হিসাব রাখতে হবে। আর স্টকের হিসাবটা রাখতে হবে শার্ট কয়টা আছে, প্যান্ট কয়টা আছে এই সিস্টেমে। সে জন্য একটা key-value ডিক্লেয়ার করা লাগবে। key হবে আইটেমের নাম আর কয়টা আছে সেই সংখ্যা হবে value। এই key-value করার জন্য stock নামে একটা প্রপার্টি ডিক্লেয়ার করছে সে।

২. sellItem নামে একটা মেথড ডিক্লেয়ার করছে। যেটার মধ্যে দুইটা ইনপুট প্যারামিটার নিচ্ছে। কোনো জিনিস কিনতে চাচ্ছে সেটার নাম (name)। আর কতগুলো কিনতে চাচ্ছে সেই পরিমাণ (quantity)

৩. কাস্টমার যে জিনিসটা কিনতে চাচ্ছে, সেটা স্টকে যথেষ্ট পরিমাণ আছে কি না চেক করার জন্য প্রথমেই স্টকে কয়টা আছে জানতে হবে। সেটা জানার জন্য একটু আগে stock নামে একটা key-value ডিকশনারি ডিক্লেয়ার করা হইছে। এখন সেই ডিকশনারিতে যে জিনিসটা কিনতে চাচ্ছে, সেটাকে key হিসেবে দিলে কয়টা আছে বলে দিবে। সেটা করার জন্য this.stock লিখে থার্ড ব্র্যাকেটের মধ্যে name লিখে দিচ্ছে। তাহলে stock নামক ডিকশনারি থেকে জানা যাবে কয়টা আছে। যে মান পাওয়া যাবে সেটা available ভেরিয়েবলে রাখছে।

৪. এখন চেক করে দেখবে কাস্টমার যতগুলো আইটেম কিনতে চাচ্ছে তার চেয়ে কম আছে কি না। আর তুলনা করার জন্য সব সময়ই if ব্যবহার করা হয়। তাই if-এর পরে ব্র্যাকেটের ভিতরে লেখা আছে available < quantity অর্থাৎ এই শর্ত সত্য হলে দোকানে যতগুলো আছে সেটা কাস্টমার যতগুলো চাইছে তার চেয়ে সংখ্যায় কম। আর কম থাকলে কাস্টমারকে সরি বলে দিচ্ছে।

৫. যদি available-এর মান quantity-এর মানের চেয়ে কম না হয়। অর্থাৎ কাস্টমার যতগুলো চাইছে তার সমান বা বেশি আইটেম স্টকে থাকলে, if-এর ভিতরে না গিয়ে else-এর ভিতরে যাবে। সেখানে প্রথমেই কাস্টমার যে জিনিস কিনতে চাচ্ছে, সেটার দাম বের করতে হবে। সেটা বের করা খুবই সিম্পল।

getPrice নামে একটা মেথড আগে থেকেই লেখা আছে। জাস্ট জিনিসের নাম দিয়ে সেই মেথডকে কল করলেই সেটার দাম দিয়ে দিবে। তারপর যে দাম পাওয়া যাবে তাকে itemPrice নামক ভেরিয়েবলে রাখছে।

৬. একটার দাম জেনে গেলে, কাস্টমার কত টাকা দিবে সেটা ক্লাস থ্রির বাচ্চাও বের করতে পারবে। জাস্ট দামের সাথে যতগুলো কিনবে, সেটার সংখ্যা দিয়ে গুণ করতে হবে। আর মোট যত টাকা দিতে হবে সেটা currentSale নামে একটা ভেরিয়েবলে রাখছে। এই currentSale হচ্ছে, এই কাস্টমারের পকেট থেকে কত টাকা খসানো হইছে সেটার পরিমাণ। তারপর এইবারের খসানোর পরিমাণ সারা দিনের টোটাল বিক্রির (totalSales) সাথে যোগ করে দিছে। যোগফলটা আবার totalSales-এর মান হিসেবেই সেট করে দিছে।

৭. সে জানে স্টকে কয়টা আছে। সেটা available ভেরিয়েবলে রাখা আছে। আবার এটাও জানে কাস্টমার কয়টা কিনতে চায়। সেটা quantity ইনপুট প্যারামিটার হিসেবে দেওয়া আছে। এখন available ভেরিয়েবল থেকে quantity ভেরিয়েবল বিয়োগ করলেই কতগুলো বাকি থাকবে জানা যাবে। এই বিয়োগফলই হবে কতগুলো বাকি আছে (remaining) ভেরিয়েবলের মান। তারপর stock নামক ডিকশনারিতে এই আইটেমের কয়টা বাকি আছে সেই value আপডেট করার জন্য this.stock লিখে থার্ড ব্র্যাকেটের ভিতরে name লিখছে। তারপর সমান চিহ্ন দিয়ে ডানপাশে remaining ভেরিয়েবল বসায় দিছে। সবশেষে কাস্টমারকে আলতুফালতু জিনিস কিনার জন্য থ্যাংকু বলে আউটপুট দেখায় দিছে।

এখন শুধু sellItem মেথডটা দেখ। এতক্ষণ সিরিয়াল ধরে যা যা পড়ছস সেটার সাথে মিলিয়ে দেখ।

```
sellItem(name, quantity){  
    var available = this.stock[name];  
    if(available < quantity){  
        console.log("Sorry we do not have enough");  
    }  
    else {  
        var itemPrice = this.getPrice(name);  
        var currentSale = itemPrice * quantity;  
        this.totalSales = this.totalSales + currentSale;  
        var remaining = available - quantity;  
        this.stock[name] = remaining;  
    }  
}
```

```

        console.log("Thank you for the purchase.");
    }
}

```

ছয়টা ফিচারের মধ্যে পাঁচটা কিন্তু করা হয়ে গেছে। তোর অ্যাপের বেশির ভাগ কাজ শেষ। আর ছোট একটা ফিচার বাকি আছে। সেটা শেষ হয়ে গেলেই কিন্তু পার্টি হবে। পার্টি।

নিত্যনতুন মাল ওঠাও, বিক্রি করে আখের গোছাও

লাস্ট ফিচারটা তেমন ভেজাইল্লা না। তা ছাড়া একটু আগে তুই addItem নামে একটা মেথড লিখছিলি। সেটাকে আরেকটু টেনেটুনে বড় করছে তোর বন্ধু ষাঁড়। সে তার addItem মেথডে তিনটা ইনপুট নিচ্ছে name, quantity, price। প্রথমটা হচ্ছে যে জিনিস ওঠানো হচ্ছে তার নাম, দ্বিতীয়টা হচ্ছে কতগুলো ওঠানো হচ্ছে। আর তৃতীয়টা হচ্ছে যে জিনিসটা ওঠানো হচ্ছে তার দাম। আর পুরা মেথডটা করতে তিনটা স্টেপ ফলো করছে সে।

১. প্রথমেই দেখতেছে জিনিসটা আছে কি না। সেটা চেক করা কোনো ব্যাপারই না। তুই isItemAvailable নামে একটা মেথড ডিক্লেয়ার করছিলি। সেটাকে কল করলে যে আউটপুট পাওয়া যাবে সেটা isExisting ভেরিয়েবলে রাখছে।

২. এখন চেক করে দেখবে isExisting-এর মান true কি না। সেই জন্য if-এর পরে isExisting == true লিখছে। যদি এই শর্ত সত্যি হয় তাহলে যে আইটেমটা ওঠানো হচ্ছে, সেটা আগে থেকেই দোকানে আছে। আর আগে কতগুলো ছিল সেটা বের করার সিস্টেম তুই আগের মেথডে লিখছিলি। এইবারও তা-ই কর। if-এর পরে সেকেন্ড ব্র্যাকেটের ভিতরে available নামে একটা ভেরিয়েবল ডিক্লেয়ার কর। যেটার মান হবে this.stock[name]। তাহলে যে জিনিসটা দোকানে ওঠাচ্ছে সেটা অলরেডি স্টকে কয়টা আছে, তা available ভেরিয়েবলে জমা হয়ে যাবে। এখন এই আইটেমটা স্টকে কতগুলো আছে সেটা আপডেট করার জন্য বামপাশে this.stock[name] লিখবি। তারপর সমান চিহ্ন দিয়ে ডানপাশে available + quantity লিখে দিবি। available মানে আগে কয়টা ছিল। আর quantity মানে এখন কয়টা ওঠাতে চাচ্ছে। এই দুইটা যোগ করে দিলে ওই আইটেমটা মোট কত পিছ দোকানে আছে তা আপডেট হয়ে যাবে।

৩. যদি আইটেমটা দোকানে আগে থেকে না থাকে তাহলে isExisting-এর মান false হবে। তখন সে if-এর পরের সেকেন্ড ব্র্যাকেটের ভিতরে যাবে না। বরং else-এর পরের সেকেন্ড ব্র্যাকেটের ভিতরে যাবে। যেহেতু নতুন আইটেম দোকানে

ওঠানো হচ্ছে তাই জিনিসটার নাম items নামক array-তে push করা লাগবে। সেটা করার জন্য this.items.push লিখে প্রথম ব্রাকেটের ভিতরের আইটেমের নামটা দিয়ে দিবি। আর জিনিসটা যেহেতু নতুন, তাই জিনিসটার দাম prices ডিকশনারিতে যোগ করা লাগবে। সেটা করার জন্য this.prices[name] লিখে সমান চিহ্ন দিয়ে ডানপাশে যে দামটা ইনপুট প্যারামিটার হিসেবে নেওয়া হইছে, সেটা বাসায় দিবি। সবশেষে, এই আইটেম যতগুলো ওঠানো হচ্ছে, সেটা stock ডিকশনারিতে যোগ করে দিবি।

সকাল-বিকাল কোডিং করি, প্র্যাকটিস করে জীবন গড়ি

তোর কোডিং করার মেইন কাজ শেষ। মনে মনে শুকরিয়া আদায় কর। এখন তোর কাজ হচ্ছে কোড ঠিকমতো কাজ করতেছে কি না, সেটা চেক করা। সেটা করার জন্য প্রথমেই www.habluderAdda.com/console-এ যাবি। তারপর পুরা ক্লাসটা একটু একটু করে লিখবি। একটা একটা মেথড করে লেখা হয়ে গেলে সেটা ঠিকমতো কাজ করতেছে কি না, চেক করবি। একটু একটু করে চেক করে করে আগাইতে থাকবি। আর নিজে নিজে কোডিং করতে না পারলে একটু পরে যে ক্লাসটা দেওয়া আছে, সেটা দেখে দেখে হলেও টাইপ করবি।

পুরা ক্লাসটা ডিক্লেয়ার করা শেষ হলে habluStore নামে একটা ভেরিয়েবল ডিক্লেয়ার করবি। যেটার মান হবে Store ক্লাস থেকে বানানো অবজেক্ট। আর Store ক্লাসের constructor ফাংশনে যেহেতু একটা ইনপুট প্যারামিটার আছে, সেহেতু Store ক্লাসকে কল করার সময় দোকানের নাম হিসেবে Bolod-Shaar Fashion House নাম দিয়ে দিবি। কারণ তুই বলদ আর তোর বন্ধু ষাঁড় মিলে এই habluStore বানাইছস। তার নিচে habluStore লিখে ডট দিয়ে addItem মেথডের নাম লিখবি। যাতে habluStore-এর addItem মেথডকে কল করা যায়। তারপর আইটেমের নাম হিসেবে দিবি shirt, পরিমাণ হিসেবে দিবি 40 আর দাম হিসেবে দিবি 300; তাহলে হাবলু স্টোরে 300 টাকা দামের 40টা শার্ট যোগ হয়ে যাবে। একই সিস্টেমে habluStore-এ 500 টাকা দামের 20টা প্যান্ট যোগ কর।

```
class Store {
  constructor(name) {
    this.name = name;
    this.items = [];
    this.stock = {};
    this.prices = {};
    this.totalSales = 0;
```

```

    }
    isItemAvailable(name){
        var itemIndex = this.items.indexOf(name);
        if(itemIndex == -1){
            return false;
        }
        else {
            return true;
        }
    }

    getPrice(name){
        var isAvailable = this.isItemAvailable(name);
        if(isAvailable == true ){
            var price = this.prices[name];
            return price;
        }
        else {
            console.log("sorry we do not have", name);
        }
    }

    getTotalSale(){
        return this.totalSales;
    }

    sellItem(name, quantity){
        var available = this.stock[name];
        if(available < quantity){
            console.log("Sorry we do not have enough");
            return;
        }
        else {
            var itemPrice = this.getPrice(name);
            var currentSale = itemPrice * quantity;
            this.totalSales = this.totalSales + currentSale;
            var remaining = available - quantity;
            this.stock[name] = remaining;
            console.log("thanks for your purchase");
        }
    }

    addItem(name, quantity, price){
        var isExisting = this.isItemAvailable(name);
        if(isExisting == true){

```



```

        var available = this.stock[name];
        this.stock[name] = available + quantity;
    }
    else {
        this.items.push(name);
        this.prices[name] = price;
        this.stock[name] = quantity;
    }
}
}
}

```

```

var habluStore = new Store("Hablu Fashion Store");
habluStore.addItem("shirt", 40, 300);
habluStore.addItem("pant", 20, 500);

```

ওপরের কোড কমপক্ষে পাঁচবার পড়ে www.habluderadda.com/console-এ গিয়ে দেখে দেখে টাইপ করে run বাটনে চাপ দিবি। run বাটনে চাপ দেওয়ার পর যদি কোনো error দেখায় তাহলে ভুল বের করার চেষ্টা করবি। ভুল খুঁজে না পেলে পেজ রিফ্রেশ দিয়ে আবার প্রথম থেকে শুরু করবি।

ঠিকমতো run করতে পারলে, স্টোরে জুতা আছে কি না চেক করার জন্য `habluStore.isItemAvailable("juta")` লিখে রান কর। দেখবি আউটপুট হিসেবে false দেখাবে। একই সিস্টেমে shirt আছে কি না, চেক করলে আউটপুট হিসেবে true দেখাবে। তারপর `habluStore.addItem("juta", 10, 200)` মেথড ব্যবহার করে 10টা জুতা স্টোরে যোগ কর। জুতা যোগ করা হয়ে গেলে `habluStore.sellItem("juta", 5)` লিখে run বাটনে চাপ দিয়ে 5টা জুতা বিক্রি করে দে। কিছুক্ষণ পরে 7টা শার্ট বিক্রি কর। তারপর `habluStore.getTotalSale()` মেথডকে কল করে সেদিন কত টাকা বিক্রি হইছে সেই হিসেব জানতে চা।

এই রকম আরো প্রোগ্রামিং প্র্যাকটিস করতে চাইলে যখন সময় পাবি তখন আড্ডা দিতে www.habluderadda.com/concepts/app-এ চলে যাবি।

প্রোগ্রামিংয়ের কোড ১০৮

ঝংকার মাহবুব



পুরা বইটা পাওয়া যাবে নীলক্ষেতের মানিক লাইব্রেরিতে (01735-742-908) এবং
রকমারিতে: <https://www.rokomari.com/book/128993>