



By Jemin Information Technology (C) 2016

## About the Author and Preface

This JSTAR is Designed by Analzing many webdesign papers  
Using JSTAR you can build any webdesign as fast as could.  
I Thank God for this wisdom given to me...

-----Wilmix Jemin J , Jemin Information Technology

This EBOOK is Printed in Asia.

To Make Software Fast like Rabbit movement

and a global redistribution of prosperity

@2016 JeminInformationTechnology , All Rights Reserved

## Acknowledgments

---

We'd like to acknowledge all of the people who played important roles in the creation of this book. We'd also like to thank all of the developers who've spent time reading this manuscript and pointing out all of the problems.

Finally, we'd like to extend a sincere thank you to the people who participated in the JSTAR Program. In particular, those who've left feedback in the Author

Online forum have had a strong impact on the quality of the final printed product.

And for providing English translations of the text resources, we'd like to thank Github and our supporters.

Thanks to all!

-----WILMIX JEMIN J

## About this Book

Welcome to JSTAR WebDesign Programming! If you've picked up this book, we suspect you're a C/C++ or PHP or JAVA or Dotnet Professional.

working with JSTAR who's somehow or other heard about webdesign like PHP, JQUERY,etc.

Perhaps you've worked with the Other Technologies in the past, perhaps you've worked with another Technologies , or perhaps this is your first step into JSTAR security with wnosql db. Whichever path has led you here, you're probably looking for a good introduction to the new JSTAR Programming Language. This book intends to give you that introduction and much more.

If you've never heard of CDOLLAR, JAVA7, we cover the basics in enough

depth to keep you in tow. If you know what JAVA7 , CDollar does, but want a deeper understanding of how it does it, we'll provide that too.

## Roadmap

Book is focused on JSTAR Programming Language , if you have knowledge or experience about JAVA and C# you can easily focus it.

But Minimum JAVA or C# Technical Knowledge is required to focus on Studying, Designing JSTAR Webdesign Modules .

JSTAR is a beautiful Programming Language focused on Webdesign and Website construction.

## *The Brief Contents*

<i>UNIT 1</i>	<i>Introduction to JSTAR Programming Language</i>	8-12
<i>UNIT 2</i>	<i>JSTAR with CDollar and Advantages over other P.L</i>	13- 15
<i>UNIT 3</i>	<i>JSTAR WebPage Syntax</i>	16 -23
<i>UNIT 4</i>	<i>JSTAR AJAX,Forms,Bills,etc Animation and Graphics , oops concepts</i>	24 - 37
<i>UNIT 5</i>	<i>XMLDATA to JSON ,RSS FEED ,WILMIXTHEME,and ANIMATION</i>	38 - 42

<i>UNIT 6</i>	<i>Jstar WITH dataStructures, Structures, with wnosql db And JSTAR Advanced Concepts</i>	43 -65
<i>UNIT 7</i>	<i>JSTAR Programmer Developer Exercises</i>	64- 81

<i>UNIT 8</i>	<i>JSTAR with Linux Platform And Overview about Jstar Syntax and Fundamentals</i>	82 - 107
<i>UNIT 9</i>	<i>JSTAR ADVANCED CONCEPTS</i>	108 - 117

<i>UNIT 10</i>	<i>JSTAR ADVANCED EXAMPLES</i>	118 - 143
<i>UNIT 11</i>	JSTAR WITH PHP , OAKJAVA7 , CDOLLAR, GDOLLAR, AND DOTNET	144 - 145
<i>UNIT -12</i>	<i>JSTAR MOCK EXERCISES</i>	146 - 148

## Code conventions

The following typographical conventions are used throughout the book:

- Courier typeface is used in all code listings.
- Courier typeface is used within text for certain code words.
- Italics are used for emphasis and to introduce new terms.
- Code annotations are used in place of inline comments in the code. These

highlight important concepts or areas of the code.

### **Code downloads**

This will get you the JSTARSERVER.zip file by downloading it.  
a couple of JSTAR archive files —as well as some documentation  
of the source. Instructions on how to install the application are contained  
in a README file in that download.

=====

## UNIT-1: Introduction to Jstar Programming Language

=====

### **Definition:**

=====

"JSTAR(J\*) is defined as a web server used for publishing  
webapplication focused on webdesign , website construction ,etc".

JSTAR(J\*) is a Programming Language which is focused on Remote webapplication in the areas of webdesign, websites construction, and which is invented in CDollar P.L.

JSTAR(J\*) is used in Remotewebapplication webdesign invented by wilmix jemin j in CDollar P.L.

### **Why you need JSTAR webdesign?**

In today world ordinary webdesign P.L programmers stores data in mysql which  
data can be easily stealed by hackers..

If you use wdbasql database with JSTAR language.

WDBA provides more security even though wdba database  
work in windows platform.



**Advantages of using JSTAR**

=====

- a) Has CDollar utilities and GDollar utilities.
- b) SAVES time and cost
- c) It requires only less source code  
to design a webdesign .
- d)Interact with WDBASQL database in Linux to provide high security.
- e) User friendly and used in windows platform.
- f) Behave Like PHP and have attractive statements.
- g) Uses CDollar-JAVA compiler and it used CDollar-java runtime
- h) accepts and executes .jstar,html , and .exe file
- i) Can be used in wallet and mobiles.
- j) solve complex problems in webdesign.
- k) Uses C# ,CDollar , GDollar , JAVA Logic and utilities.

l) it is used with WEB1.0 webservice ..

m) JSTAR is focused on remote webdesign without internet connection is the major advantage

of JSTAR.

n) JSTAR is platform independant

o) JSTAR is Port independant (ie; Port is expected to change everytime when you run the webserver.

(this is another major advantage of jstar and hence maintains security)

### **What is JSTAR ? Why it is focused?**

Jstar is a webdesign Programming Language(P.L) invented by wilmix jemin j in year 2014.

JStar uses Wilmix remote Server to publish the Webpage directly in the internet..

So Users can see the webpage from internet ,mobiles,and ,WALLET..

So using this P.L ,client can access the webpage through WALLET, mobiles

and other computers, laptop,et. And the (webdesign JSTAR Program) will even travel in web...

### **HISTORY of JSTAR**

JSTAR is invented in CDollar P.L @ year 2014 by wilmix jemin j.

we can easily build a JSTAR Program...

**SYNTAX:**

<JSTAR>

<LOGIC>

<! JSTAR LOGIC !>

</LOGIC>

</JSTAR>

Note: JSTAR follows CDollar Programming style

**How JSTAR Technology Works? Or JSTAR workflow?**

---

At first .jstar is compiled by remoteserver using CDollar -CDC compiler.

And which translates to (bytecode) .class file .

This JAVAC compiler is divided into two parts

a) accept corejava concepts

b) accept CDollar concepts

remoteserver also executes .exe file.

so java programmers can use it.

shortly when client send a request to a remoteserver ;

remoteserver accepts JSTAR code and process the request

and print the output generated by CDC compiler by

using JSTAR JAVA runtime.

And JSTAR server sees any content from <html> and

</html> TAGS. So make sure at all program should start with

<html> .... And and end with </html> tag.

How to run JSTAR remoteserver in windows?

click JSTARSERVER.exe and make the path

point where JSTARSERVER.exe files resides.

How to run JSTAR remote server in Linux type os?

enable read write permission of file JSTARSERVER1

and run using...

./JSTARSEVER1

Note: In JSTAR we use OAKJAVA(JAVA7) to make the page as fast.

=====

## UNIT -2: a) JSTAR with CDollar And Advantages over other P.L.

=====

### A) CDollar with JSTAR OVERVIEW

We already know that CDollar p1 (CWE -EDITOR) or CDollarcc is used for construction of .exe prototype and .dll files.

So if you use CDollar with JSTAR then CDollar => .exe and .dll will be obsucated when compared to other prototype files.

CDollar is used with Advanced oops than can make the program work faster.

Since such datastructures are made by pointers. Hence CDollar Program

Execution will be faster than other Technologies.

### **JSTAR Advantages over JAVA and other Programming Languages**

-----

a) Has CDollar utilities and GDollar utilities.

b) SAVES time and cost

c) It requires only less source code

to design a webdesign .

d) User friendly and used in windows platform.

e) Behave Like PHP and have attractive statements.

f) Uses CDollar-JAVA compiler and it used CDollar-java runtime

g) accepts and executes .jstar,html , and .exe file

h) Can be used in wallet and mobiles.

i) solve complex problems in webdesign.

j) Uses C# ,CDollar , GDollar , JAVA Logic and utilities.

k) it is used with WEB1.0 webservice ..

l) JSTAR is focused on remote webdesign without internet connection is the major advantage of JSTAR.

m) JSTAR is Port independant (ie; Port is expected to change everytime when you run the webserver.

(this is another major advantage of jstar and hence maintains security)

n) JSTAR Generated only .class file. ie) it equivalent to OAKJAVA7.

since .class file is faster than using CDollar compiler

and .exe is more faster than .class file.

o) JSTAR is platform independant and advanced than php technology.

**FAQS**

-----

A) A C /C++ Programmer or any oops developer or java7(OAKJAVA) programmer  
can easily study it....

**What is difference between webserver and remotewebserver?**

Webserver will not be published directly to mobiles.

it requires Internet permission and authentication.

but on the other had remoteserver will host the webpages

in the internet and publish directly to mobiles..

so remoteserver is the hosting server ...

JSTAR server is a remote server.

### UNIT-3: JSTAR Remote Webpage Syntax , JSTAR7.0 (similar to oakjava7)

SYNTAX to execute executable files by remoteserver:

```
RuntimeExec.call("EXECUTABLEFILENAME")
```

What is the syntax to get and splitting the parameters ?

```
<AList> <String> armg=Request.Query(arraylistobject,"arraylist.dsn",howmanyGUI,incrementor);
```

```
Displayhtml.Design1("http://www.hdwallpaperspulse.com/wp-content/uploads/2013/08/08/wallpaper-115621.jpg",250,300); => load this wallpage in jstar webpage
```

```
Displayhtml.Design2(); => apply design2
```

```
$HTML.displayhtml("prog.html"); => apply canvas or html designs
```



\$HTML.AnimateHtml("prog2.html"); => apply svg

\$HTML.displayhtml("r.html"); => apply jquery with JSTAR AJAX

\$HTML.displayhtml("animate.html"); => perform animation

\$XMLDATA.TOJSON("rss2.xml"); => convert to json format

\$ParseXMLDOM.Process("http://news.google.com/?output=rss"); => convert to rss feed

JTHEMES.WILMIXTHEME1("MOBILE THEME" , "" , "weldoen", "weldone" , "" , "DDSdS" , "dsdsds" , "dsdsdsds" ); => apply mobile theme

\$HTML.AnimateHtml("chart2.html"); => perform animation

JSTAR.WriteLine("printstring") =>

It is used to print the string or value to the webpage

EXAMPLE-1:

=====

Create a remote webpage with register page containing 3 text box

and print the value in table format using jstar.

Register.html

```
<html>

<head>

  <title> USER REGISTRATION</title>

</head>

<body class="fancy">
<form action="http://localhost:8090/Program5.exe" method="post" >

<div id="pageContainer">

  <div id="pageContent">

    <div id="chaptersAccordion">

      <h2><a href="#chapter1">Enter your System Details</a></h2>

      <div>

<p>Enter the Name: <input type="text" name="name" size="25" /></p>
<p>Enter your Username : <input type="text" name="uname" size="15" /></p>
<p>Enter the password : <input type="password" name="password" size="25" /></p>

      </div>

    </div>

  </div>

</div>
```

```
<h2><a href="#chapter2">REGISTER</a></h2>
```

```
    <div>
```

```
        <input type="submit" name="Click">
```

```
<input type="reset" name="Clear">
```

```
    </div>
```

```
</form>
```

```
</body>
```

```
</html>
```

**index.jstar**

```
<JSTAR>
```

```
<LOGIC>
```

```
public class index
```

```
{
```

```
public void main( )  
{  
  
$HTML.displayhtml("Register.html");  
  
}  
  
}
```

</LOGIC>

</JSTAR>

### **JSTAR7.0 Program**

<JSTAR7>

<USE> <CDOLLARS>.util;

<PACK> Program5

```
{

    <CLASS> Prog
    {

        public void main()
        {

            ArrayList arm1 = new ArrayList();

            arm1.add("name");
            arm1.add("uname");arm1.add("password");

            arm1.add("NOT");


            ArrayList armg= Request.Query(arm1,"index.dsn",3,1);

            // choose as index.dsn the page you input
            //and if there are 3 textboxes then increment counter =1


            //write the html tables tag in println statement
```

```
<PRINTLN>("<table style='width:100%' bgcolor=gold>");
```

```
<PRINTLN>("<tr>");
```

```
<PRINTLN>(" <th>Name</th>");
```

```
<PRINTLN>(" <th>Username</th>");
```

```
<PRINTLN>(" <th>Password</th>");
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>(" <tr>");
```

```
for (int i=0;i<armg.size();i++)
```

```
{
```

```
<PRINTLN>("<td>"+armg.get(i)+"</td>");
```

```
//get the values from arraylist and print those values in table data
```

```
}
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>("</table>");
```

```
}
```

```
}
```

```
}
```

=====

## UNIT-4:

=====

JSTAR AJAX ,JSTAR Forms , Bills,etc,Animation and Graphics  
 ,oops concepts

=====

### JSTAR SERVLET

JSTAR.WriteLine("string");

this above statement will print the string on the webpage will  
 act like JSTAR servlet

### JSTAR AJAX

JSTAR.WriteLine("<form action=BILL.jstar method=post>");

if BILL.jstar is the current JSTAR program when you give  
 method=post and action=BILL.jstar it will act like JSTAR AJAX.

### JSTAR WXML METHODS

#### SYNTAX:

WXML.wxml<PARSE>r("filename.wxml", wxmlindex,string+""");

for example)

WXML.wxml<PARSE>r("bill.wxml", 4,s3+""");

=> this statement is used to print bill , Forms,TABLES.



4=> indicates bill format

1=> indicates Forms

2=> indicates TABLES

For Animation and Graphics use the following statements

\$HTML.displayhtml("prog.html"); => apply canvas or html designs

\$HTML.AnimateHtml("prog2.html"); => apply svg

JSTAR MVC

=====

JSTAR MVC is userdefined it is very easy for developers

to focus like struts.

take 3 class Model(wdba database code) , formview (with set and get methods),

controller .

call controller class result to jstar main program

ie)

```
public void main( ) throws <EXE>
```

```
{
```

```
<AList><String> arm1= <NEW> <AList><String>();

arm1.add("name");

arm1.add("uname");arm1.add("password");

//assume 3 textbox fields with name ,uname,password

arm1.add("!");

<AList><String> armg=Request.Query(arm1,"Jquerytest.dsn",3,1);

//this statements are mandaltory eg)

//to get the value from WeBGui like text box ,etc.

controller.set(name); //set form values.

JSTAR.WriteLine(controller.get());

//this statement will print the output in webpage

}
```

## Program-1:JSTAR program with Thread and oops concepts

=====

&lt;JSTAR&gt;

&lt;USE&gt; &lt;J&gt;.util.\*; //load jstar util packages

&lt;LOGIC&gt; =&gt; Logic of JSTAR program

abstract class abcd &lt;--- &lt;THREAD&gt;

{

&lt;VOLATILE&gt; int v1;

Shared &lt;TRANS&gt; int v;

~

{

JSTAR.WriteLine("object is deleted"); //print the string to webpage

}

public int getnum1()

{

```
return(v1);  
  
}  
  
Shared int s3;  
  
public Shared void operator *(int s1 ,int s2) //operator overloading  
{  
  
    s3=s1 * s2;  
  
    JSTAR.WriteLine(""+s3);  
}  
  
public void main( ) throws <EXE>  
{  
  
    int a[] <Array> int[1000]; //declaring array  
  
    operator *(10,10);  
  
    operator *(200,10000);  
  
    JSTAR.WriteLine("<html>");
```

```
<AList> ar = <NEW> <AList>();
```

```
for(int i=999;i>=0;i--)
```

```
{
```

```
    a[i]=i;
```

```
    ar.add(i);
```

```
}
```

```
<A>.<SORT>(a); //soriting the array contents
```

```
for(int i=0;i<=999;i++)
```

```
    JSTAR.WriteLine(" "+i);
```

```
    JSTAR.WriteLine(" "+<A>.<BinarySearch>(a,-5));
```

```
<WR> it = ar.record();
```

```
while(it.<HAS>) // simillar to it.hasNext in java
```

```

{
<OBJECT> el = it.<NEXT>;

JSTAR.WriteLine(" data= "+el);

}

<LR> it1 = ar.<LISTLR>;
while(it1.<HASP>)

{
<OBJECT> el1 = it1.<PREVIOUS>; //move to previous point

JSTAR.WriteLine(" data1= "+el1);
}

<-----
while (es.<HASEMORE>)

{

<OBJECT> el11 = es.<NEXTEL>;

JSTAR.WriteLine(" data1= "+el1);
}

```

----->

```
<DATE> d2 = <NEW> <DATE>();
```

```
JSTAR.WriteLine("month="+<Month>);
```

```
JSTAR.WriteLine("Year="+<Y>);
```

```
JSTAR.WriteLine("Hour="+<H>);
```

```
JSTAR.WriteLine("Sec="+<SEC>);
```

```
//<VOLATILE> int v1;
```

```
//<TRANS> int v;
```

```
v=20;
```

```
<FOUTPUTSTREAM> os1 = <NEW> <FOUTPUTSTREAM>("out1.txt");
```

```
<OOUTPUTSTREAM> d = <NEW> <OOUTPUTSTREAM>(os1);
```

```
d.<WRITEOBJ>(v);
```

```
d.<WRITEOBJ>(ar);
```

```
<FINPUTSTREAM> osd = <NEW> <FINPUTSTREAM>("out1.txt");
```

```
<OINPUTSTREAM> br = <NEW> <OINPUTSTREAM>(osd);
```

```
<OBJECT> ar7 = br.<OBJECTREAD>;
```

```
JSTAR.WriteLine("""+ar7);
```

```
<STACK> s= <NEW> <STACK>();
```

```
s.<PUSH>(100);
```

```
s.<PUSH>(2000);
```

```
s.<POP>;
```

```
JSTAR.WriteLine("""+s);
```

```
<PRIORITYQUEUE> q = <NEW> <PRIORITYQUEUE>();
```

```
q.add(1000);
```

```
q.add(544);
```

```
q.add(66);
```

```
q.add(667888);
```

```
JSTAR.WriteLine("""+q);
```

```
JSTAR.WriteLine("</html>");
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```



Output:

```

1002000000 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123
124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148
149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173
174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248
249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273
274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298
299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348
349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373
374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398
399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423
424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448
449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473
474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498
499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523
524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548
549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598
599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623
624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648
649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673
674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698
699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723
724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748
749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773
774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798
799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823
824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848
849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873
874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898
899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923
924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948
949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973
974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998
999 -1 data= 999 data= 998 data= 997 data= 996 data= 995 data= 994 data= 993 data= 992 data= 991 data= 990 data=

```

[illegible]

[illegible]

32 data= 31 data= 30 data= 29 data= 28 data= 27 data= 26 data= 25 data= 24 data= 23 data= 22 data= 21 data= 20 data= 19 data= 18 data= 17 data= 16 data= 15 data= 14 data= 13 data= 12 data= 11 data= 10 data= 9 data= 8 data= 7 data= 6 data= 5 data= 4 data= 3 data= 2 data= 1 data= 0 month=2 Year=1 Hour=10 Sec=1320 [100][66, 1000, 544, 667888]

### **Program-2:**

<JSTAR>

<LOGIC>

<USE> GRAPHICS.\*; //load graphics packages

public class abc

{

public void main( )

{

Displayhtml.Design1("http://www.hdwallpaperspulse.com/wp-content/uploads/2013/08/08/wallpaper-115621.jpg",250,300); //load this wallpage in

//jstar webpage

Displayhtml.Design2(); //apply design2

\$HTML.displayhtml("prog.html"); //apply canvas

```
$HTML.AnimateHtml("prog2.html");//apply svg
```

```
$HTML.displayhtml("r.html");//apply jquery with JSTAR AJAX
```

```
$HTML.displayhtml("animate.html");//perform animation
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```

```
=====
```

**UNIT-5:**

```
=====
```

**XMLDATA to JSON , RSS FEED, WILMIXTHEME , and Animation**

```
=====
```

### **Program-3**

```
=====
```

```
<JSTAR>
```

```
<LOGIC>
```

```
public class Chart
```

```
{
```

```
public void main( )
```

```
{
```

```
JSTAR.WriteLine("<html>");
```

```
$XMLDATA.TOJSON("rss2.xml");//convert to json format
```

```
$ParseXMLDOM.Process("http://news.google.com/?output=rss");//convert to rss feed
```

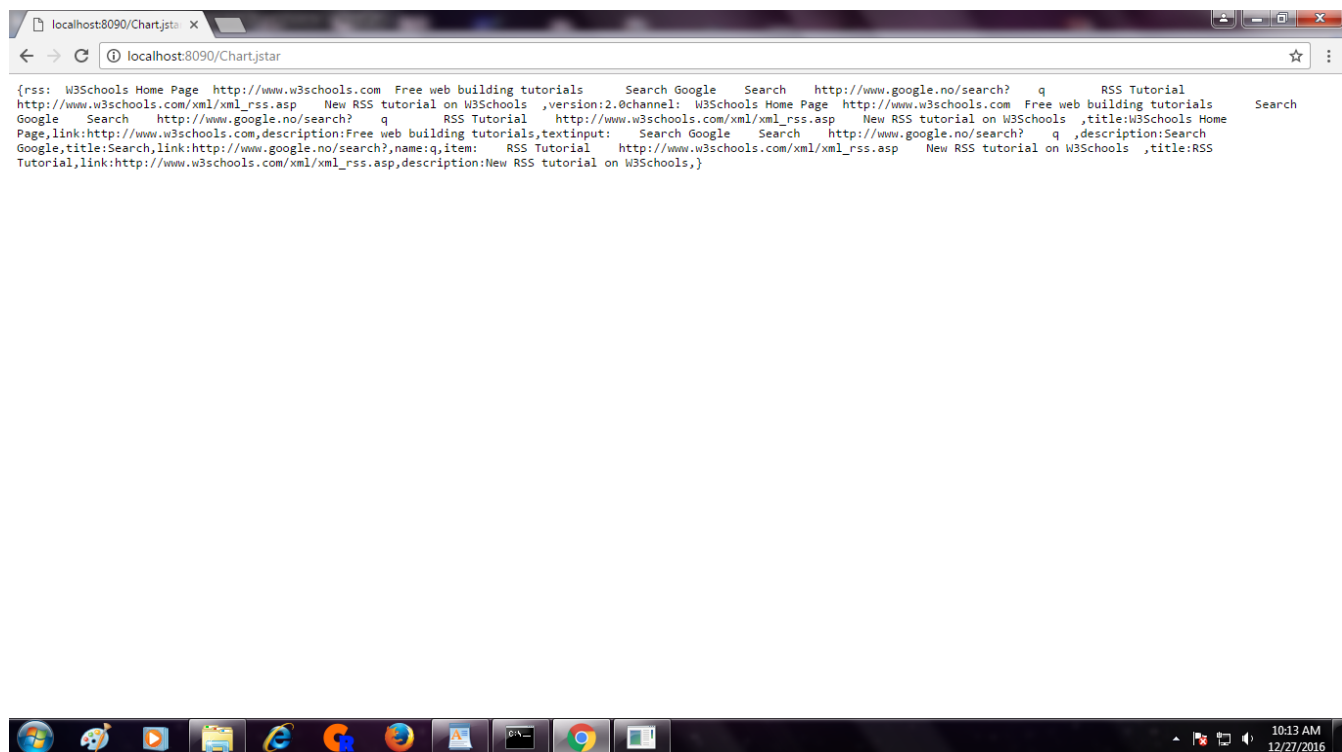
```
JSTAR.WriteLine("</html>");
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```



Program-4:

=====

chart2.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#svgelem {
```

```
    position: relative;
```

```
    left: 50%;
```

```
    -webkit-transform: translateX(-20%);
```

```
    -ms-transform: translateX(-20%);
```

```
    transform: translateX(-20%);
```

```
}
```

```
</style>
```

```
<title>SVG</title>
```

```
<meta charset="utf-8" />
```

```
</head>
```

```
<body>
```

```
<h2 align="center">HTML5 SVG Circle</h2>
```



```
<svg id="svgelem" height="200" xmlns="http://www.w3.org/2000/svg">
```

```
  <circle id="redcircle" cx="50" cy="50" r="50" fill="red" />
```

```
</svg>
```

```
</body>
```

```
</html>/>
```

#### Program 4:2

```
<JSTAR>
```

```
<LOGIC>
```

```
public class chart2
```

```
{
```

```
public void main( )
```

```
{
```

```
JTHEMES.WILMIXTHEME1("MOBILE THEME" ,"" ,"weldoen","weldone","", "DDSdS","dsdsds","dsdsdsds" ); //apply  
mobile theme
```

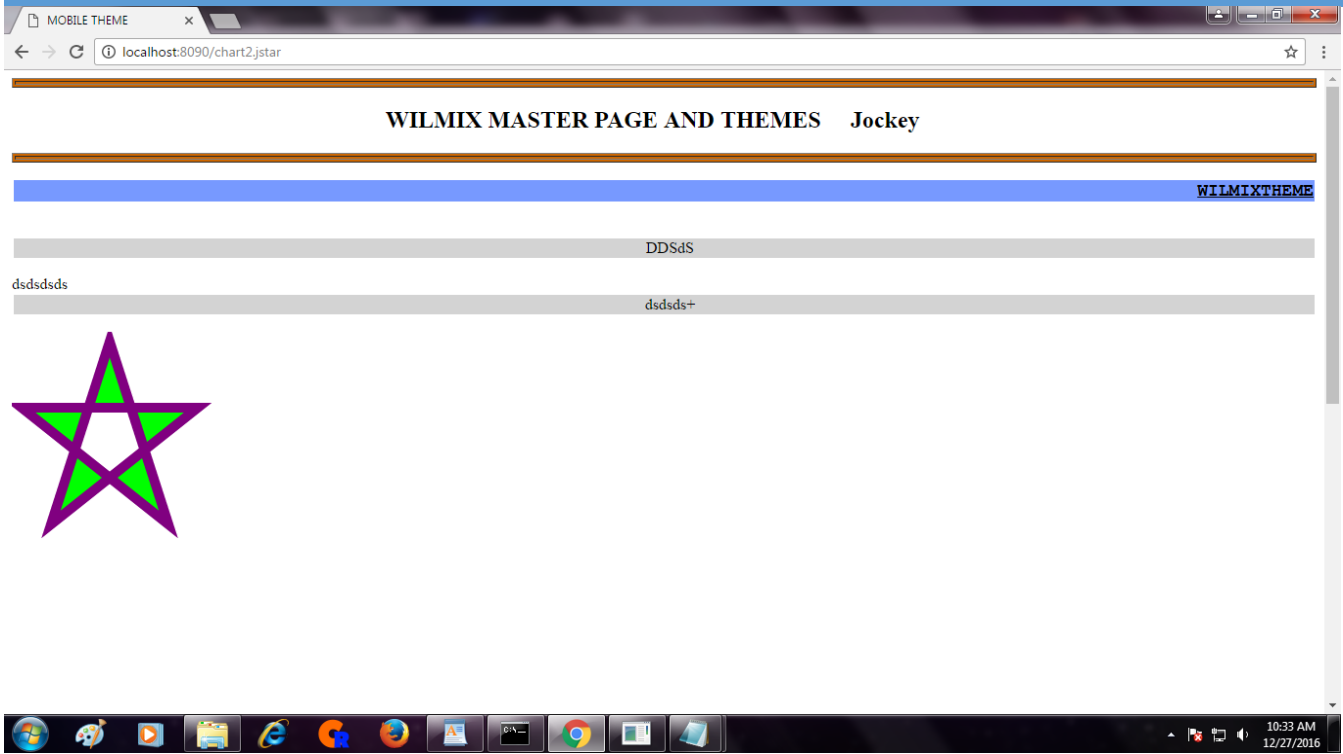
```
$HTML.AnimateHtml("chart2.html"); // perform animation
```

```
}
```

```
}
```

Output:

```
=====
```



```
=====
```

**UNIT-6:**

```
=====
```

**JSTAR with Datastructures , Structures, JSTAR with WNO SQL  
Db and JSTAR Advanced Concepts**

```
=====
```

**Program-5:JSTAR with Linked List**

<JSTAR>

<LOGIC>

```
//a node in a linked list
class LList
{
    //the next node in the list
    private LList nextNode = null;
```

```
//the datum stored in this node
```

```
private String datum = null;
```

```
public LList(String datum)
```

```
{
```

```
    <SUPER>();
```

```
    <IS>.datum = datum;
```

```
}
```

```
public void add(String datum)
```

```
{
```

```
    if (nextNode NOT= null)
```

```
    {
```

```
        nextNode.add(datum);
```

```
        return;
```

```
    }
```

```
    nextNode = <NEW> LList(datum);
```

```
}
```

```
public <Str> get(int i)
```

```
{
```

```
    if (i == 0)
```

```
        return datum;
```

```
    return nextNode.get(i - 1);
```

```
}
```

```

public int size()
{
    if (nextNode == null)
        return 1;

    return nextNode.size() + 1;
}

```

```

public void main( ) throws <EXE>
{
    //construct the list
    LList list = <NEW> LList("0 computer");
    list.add("1 mouse");
    list.add("2 cpu");
    list.add("3 keyboard");
    list.add("4 monitor");
    list.add("5 Usbdrive");
    list.add("6 harddisk");

    JSTAR.WriteLine("<html>");

    //print the list
    for (int i = 0; i NOT= list.size(); i = i + 1)
    {
        JSTAR.WriteLine(list.get(i));
    }
}

```

```
JSTAR.WriteLine("\n");
```

```
}
```

```
JSTAR.WriteLine("</html>");
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```

### **Program -6: Using Structures**

```
<JSTAR>
```

```
<LOGIC>
```

```
class Main  
46
```

```
{  
  
    public Main()  
    {  
        <SUPER>();  
    }  
  
    public void main1()  
    {  
  
        String s="dsdds";  
  
        {*<<Str>> I Pointer<<Str>>(s);  
  
l.add(s);  
  
        for (int i = 0; i NOT= l.size(); i = i + 1)  
        {  
  
            <OBJECT> obj=l.get(i);  
            JSTAR.WriteLine(obj);  
  
        }  
    }  
}
```

```
<STRUCTURE> <<OBJECT>> list = <NEW> <STRUCTURE> <<OBJECT>>(l.get(0).StringConvert());
```

```
for (int i11 = 0; i11 NOT= list.size(); i11 = i11 + 1)
```

```
{
```

```
<OBJECT> el=list.ret(i11);
```

```
JSTAR.WriteLine("SNO= "+el.StringConvert());
```

```
}
```

```
}
```

```
}
```

```
class wabc
```

```
{
```

```
public void main( )
```



```
{  
  
JSTAR.WriteLine("<html>");  
  
  
Main m = <NEW> Main();  
  
m.main1();  
  
JSTAR.WriteLine("</html>");  
  
  
}  
  
}  
  
  
</LOGIC>  
</JSTAR>
```

Program-8: Programs6.web

<JSTAR7>

<USE> Security; //load security packages

<PACK> Program6

{

<CLASS> Prog

{

public void main()

{

wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds",  
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod"); //give the path

//database name

wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78"); //usertable,pwdtable,user,pwd

String q="SELECTRVAL from Telecom 3 to 33 , 1 to 7 ?= A By 1 1 : {0} : {0} :{0}"; //list all the rows from 3 to 33  
from Telecom table.

wdbaconn.WDBAQUERY(q); //execute the wdbaquery

```
<TRY>
```

```
{
```

```
<PRINTLN>("" + Secure.RetrieveSecure("output.wdba", 0)); //retrieve the query output from wdba file
```

```
}
```

```
<CATCH>(<EXE> e)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
}
```

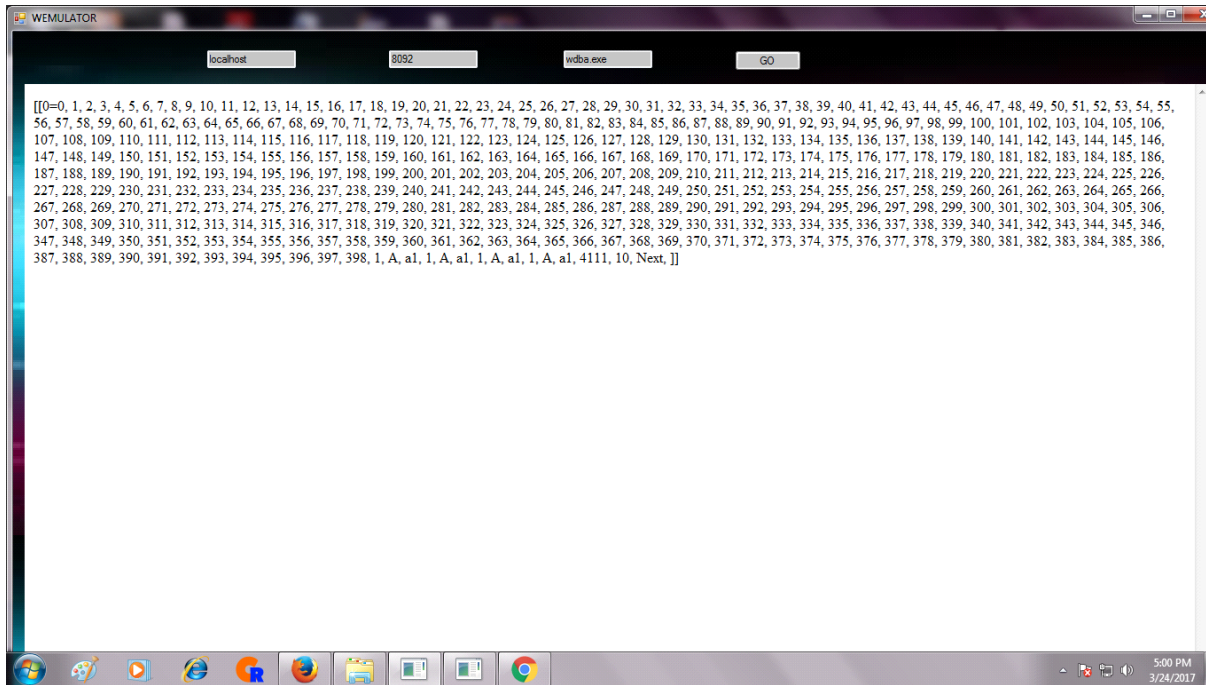
**How to compile oakJAVA7 program ?**

jstarc Programs6.web

when compiling it produces .exe file ie) Programs6.exe

rename it as wdba.exe , put where JSTARSERVER resides and execute in WEMULATOR.

### OUTPUT:



### What is the advantage of using OAKJAVA or JSTAR7?

make the page fast to display.

### Program -7 : JSTAR WITH WNOSQL

#### WNOSQL PROGRAM

<WNOSQL>

<PACK>

52

```
<USE> CDollar.WDBA;
```

```
<USE> WDBA;
```

```
<DATA LIB> ps
```

```
<DATA>
```

```
public <CLASS> DATA
```

```
{
```

```
public void main()
```

```
{
```

```
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",  
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQ");
```

```
// give database name datastores and with path
```

```
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQ"
```

```
String t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);
```

```
//supply user with wilmix78 and pwd with wilmix78
```

```
String s1 = "CREATETABLE from Employee 0 to 0 , 1 to 6 ?= X By 1 1 : {ENO,ENAME,DESIGNATION,SALARY}: {0} :{0}";
```

```
// create table with fields eno,ename,designation,salary
```

```
ArrayList ar= WDBALIB.WDBAQUERY(s1, t);
```

```
String s11 = "SELECTRVAL from Employee 0 to 0 , 1 to 6 ?= X By 1 1 : {0} : {0} :{0}";
```

```
WDBA.writeln(""+WDBALIB.WDBAQUERY(s11, t));
```

```
//list all the details of employee
```

```
String s12="INSERTINTO from Employee 4 to "+ar.size() +" , 1 to 6 ?= X By 1 1 : {0} :  
{2,jemin,sse2,300000,3,shyam,mgr,500000} : {0}";
```

```
WDBA.writeln(""+ WDBALIB.WDBAQUERY(s12, t));
```

```
//insert in to employee
```

```
String s112="SELECTRVAL from Employee 0 to 0 , 1 to 6 ?= X By 1 1 : {0} : {0} :{0}";
```

```
WDBA.writeln(""+WDBALIB.WDBAQUERY(s112, t));
```

```
// list all the details of employee
```

```
String s171="UPDATE from Employee 0 to 0 , 1 to 6 ?= C By 1 1 : {shyam} :{sriram}:{0}";
```

```
WDBA.writeln(""+WDBALIB.WDBAQUERY(s171, t));
```

```
// update the data shyam to sriram
```

```
Char c='';
```

```
ArrayList arhd1gy1178=WDBASQL.Query("MATCH","Employee","0",null,7,"mgr"," ",null,"",0,"1","",c,null,t,1,6);
```

```
//display the matched data mgr locations
```

```
WDBA.writeln(""+arhd1gy1178);
```

```
ArrayList arhd1gy=WDBASQL.Query("SelectRange","Employee",6+"" ,null,7,""," ",null,"",0,"1"," ",c,null,t,1,6);
```

```
//display details from 6 to 7
```

```
WDBA.writeln(""+arhd1gy);
```

```
}
```

```
}
```

```
}
```

## JSTAR ADVANCED CONCEPTS

```
=====
```

JSTAR also follow wizard form controller like Spring framework do.

JSTAR follows to store all the incoming parameters in .dsn file and there is

no need for .xml file . Thus it simplifies the code. This is known as JSTAR Spring.

JSTAR also follows MVC design pattern...

JSTAR generates .wl files and thus it protect from hackers.

Now Let us see more about JSTAR wizard form controller with HTML and bootstrap template.

**Design two forms one for entering father details and another for entering spouse details and print**

**those details use only jstar program and also use HTML with bootstrap template ...?**

form.jstar

=====

<JSTAR>

<LOGIC>

public class form

{

public void main( )

{

\$HTML.displayhtml("formn.html");

}



```
}
```

```
</LOGIC>
```

```
</JSTAR>
```

```
form1.jstar
```

```
=====
```

```
<JSTAR>
```

```
<LOGIC>
```

```
public class form1
```

```
{
```

```
public void main( )
```

```
{
```

```
$HTML.displayhtml("UserForm1.html");
```

```
}
```

```
}
```

</LOGIC>

</JSTAR>

UserForm1.html

=====

<html>

<body>

<head>

<title>USERFORM</title>

</head>

<form name=f1 action="Jquerytest3.exe" method="post" >

<p>Your Name : <input type="text" name="name" /></p>

<p>Your Age : <input type="text" name="age" /></p>

<input type="submit" value="Add User" />

</form>

</body>

</html>

**note: include file Jquerytest3.exe which is created using JSTAR7 or OAKJAVA7 program..**

jquerytest3.web : OAKJAVA program

=====

<JSTAR7>

<USE> <CDOLLARS>.util;

<PACK> JQuerytest36

{

    <CLASS> Prog

{

    public void main()

{

    ArrayList arm1 = new ArrayList();

    arm1.add("names");

    arm1.add("ages");

    arm1.add("name");

    arm1.add("age");

    arm1.add("NOT");

    ArrayList armg= Request.Query(arm1,"Jquerytest3.dsn",4,1);

```
<PRINTLN>("<table style='width:100%' bgcolor=gold>");
```

```
<PRINTLN>("<tr>");
```

```
<PRINTLN>(" <th>Father NAME</th>");
```

```
<PRINTLN>(" <th>Age</th>");
```

```
<PRINTLN>(" <th>NAME</th>");
```

```
<PRINTLN>(" <th>Age</th>");
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>(" <tr>");
```

```
for (int i=0;i<armg.size()-1;i++)
```

```
{
```

```
<PRINTLN>("<td>"+armg.get(i).replace("&name","").replace("names","").replace("name","")+ "</td>");
```

```
}
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>("</table>");
```

```
}  
}  
}
```

head&gt;

Logo

Home

About

Projects

Contact



Another Image Ma

More Sell \$

Lorem ipsum...

Father Name : Father Age : 

localhost:8092/form.jstar#myCarousel



## JSTAR Advanced Applications in webdesign

=====

JSTAR CAN be used in most webdesign areas like Blogging ,Capatche,

AJAX,Site Search,Integerating Google maps, and involved in youtube publication.

=====

## UNIT-7 :

=====

### JSTAR Program DEVELOPER EXERCISES

=====

#### Program-1:Write a JStar program to perform banking transctions ?

##### Transcations.html

```
<html>

<head>

<title> Banking Transcations</title>


</head>

<body class="fancy">

<form action="http://localhost:8092/Programs7.exe" method="post" >


<div id="pageContainer">


<div id="pageContent">


<div id="chaptersAccordion">
```



```
<h2><a href="#chapter1">Enter your Transcation Details</a></h2>
```

```
<div>
```

```
<p>Enter a Amt: <input type="text" name="name" size="25" /></p>
```

```
<p>Enter to Credit a Amt: <input type="text" name="credit" size="25" /></p>
```

```
<p>Withdraw of a cash : <input type="text" name="withdraw" size="25" /></p>
```

```
</div>
```

```
<h2><a href="#chapter2">Credit now</a></h2>
```

```
<div>
```

```
<input type="submit" name="CREDIT">
```

```
</div>
```

```
</form>
```

```
</body>
```

</html>

Transcation.jstar

<JSTAR>

<LOGIC>

public class Transcation

{

public void main( )

{

\$HTML.displayhtml("Transcations.html");

}

}

```
</LOGIC>
```

```
</JSTAR>
```

Write a JSTAR7 program Programs7.web for that and use javac Programs7.java to compile

so it produces .exe ie) Programs7.exe

Programs7.web

```
<JSTAR7>
```

```
<USE> <CDOLLARS>.util;
```

```
<PACK> Program7
```

```
{
```

```
    <CLASS> Prog
```

```
    {
```

```
Shared int tamt=0;
```

```
    public void main()
```

```
    {
```

```
ArrayList arm1 = new ArrayList();
```

```
arm1.add("name");
```

```
arm1.add("credit");arm1.add("withdraw");
```

```
arm1.add("!");
```

```
ArrayList armg= Request.Query(arm1,"Transcation.dsn",3,1);
```

```
<PRINTLN>("<table style='width:100%' bgcolor=gold>");
```

```
<PRINTLN>("<tr>");
```

```
<PRINTLN>(" <th>Amt</th>");
```

```
<PRINTLN>(" <th>CRedit/Debit</th>");
```

```
<PRINTLN>(" <th>Withdraw/Discount</th>");
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>(" <tr>");
```

```
tamt=Convert.ToInt32(armg.get(0).ToString())+Convert.ToInt32(armg.get(1).ToString())-  
Convert.ToInt32(armg.get(2).ToString())-Convert.ToInt32(armg.get(0).ToString());
```

```
<PRINTLN>("CREDITAMT="+tamt+"</td>");
```

```
for (int i=0;i<armg.size();i++)
```

```
{
```

```
<PRINTLN>(" <td>"+armg.get(i)+"</td>");
```

```
}
```

```
<PRINTLN>(" </tr>");
```

```
<PRINTLN>(" </table>");
```

```
}  
}  
}
```

Write a Jstar Program to insert all emp details in wnosql db and list it

<JSTAR>

<LOGIC>

```
public class index1
```

```
{
```

```
public void main( )
```

```
{
```

```
$HTML.displayhtml("Eregister.html");
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```

Eregister.html

```
<html>
```

```
<head>
```

```
<title> USER REGISTRATION</title>
```

```
</head>
```

```
<body class="fancy">
```

```
<form action="http://localhost:8092/Programs8.exe" method="post" >
```

```
<div id="pageContainer">
```

```
<div id="pageContent">
```

```
<div id="chaptersAccordion">
```

```
<h2><a href="#chapter1">Enter your EMP Details</a></h2>
```

```
<div>
```

```
<p>Enter the Name: <input type="text" name="ename" size="25" /></p>
```

```
<p>Enter your Username : <input type="text" name="dept" size="15" /></p>
```

```
<p>Enter the password : <input type="password" name="design" size="25" /></p>
```

```
</div>
```

```
<h2><a href="#chapter2">REGISTER</a></h2>
```

```
<div>
```

```
<input type="submit" name="Click">
```

```
<input type="reset" name="Clear">
```

```
</div>
```

```
</form>
```

```
</body>
```



</html>

**Write a OAKjava program Programs8.web for that and use javac Programs8.java to compile so it produces .exe ie) Programs8.exe**

<WEB>

<USE> <CDOLLARS>.util;

<USE> Security;

<PACK> Program8

{

<CLASS> Prog

{

public void main()

{

String t="";

```
ArrayList arm1 = new ArrayList();
```

```
arm1.add("ename");
```

```
arm1.add("dept");arm1.add("design");
```

```
arm1.add("!");
```

```
ArrayList armg= Request.Query(arm1,"index1.dsn",3,1);
```

```
wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds", "C:\\Prog\\WNOSQL\\WNOSQL\\WNOSQL-cod");
```

```
wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78");
```

```
String q = "CREATETABLE from EMP 0 to 0 , 1 to 5 ?= 6639 By 6639 f(x) : {EMPNAME,DEPT,DESIGNATION}: {} :{2,4}";
```

```
wdbaconn.WDBAQUERY(q);
```

```
String qh="SELECTRVAL from EMP 3 to 33 , 1 to 5 ?= A By 1 1 : {0} : {0} :{0}";
```

```
wdbaconn.WDBAQUERY(qh);
```

```
t=armg.get(0).ToString()+","+armg.get(1).ToString()+","+armg.get(1).ToString();
```

```
String s12="INSERTINTO from EMP 3 to "+3+" , 1 to 5 ?= A By 1 1 : {0} : {"+t+"} : {0}";
```

```
wdbaconn.WDBAQUERY(s12);
```

```
String qh2="SELECTRVAL from EMP 3 to 6 , 1 to 5 ?= A By 1 1 : {0} : {0} :{0}";
```

```
wdbaconn.WDBAQUERY(qh2);
```

```
<TRY>
```

```
{
```

```
<PRINTLN>(""+Secure.RetrieveSecure("output.wdba",0)); //retrieve the query output from wdba file
```

```
}
```

```
<CATCH>(<EXE> e)
```

```
{
```

```
}
```

```
}
```

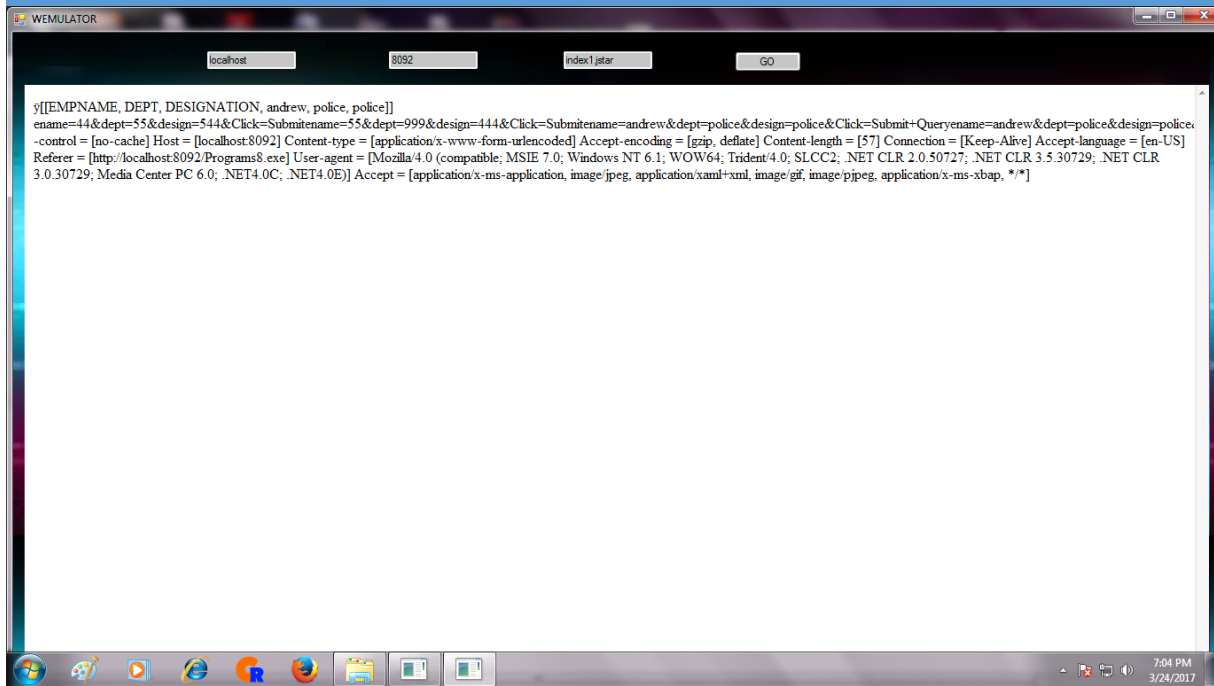
```
}
```

```
}
```

**Note:**

Now we notice that we enter details in index1.jstar

and it is stored in wnosql db and displayed in EMULATOR



## Report.wxml

<!DOCTYPE WXML [

<!ELEMENT WXML (REPORT)\*>

<!ELEMENT REPORT EMPTY>

```
//declare all the definitons needed for report
```

```
<!ATTLIST REPORT
```

```
    TITLE CDATA #IMPLIED
```

```
    Method CDATA #IMPLIED
```

```
    Uri CDATA #IMPLIED
```

```
    color CDATA #IMPLIED
```

```
    Tcolor CDATA #IMPLIED
```

```
    Name CDATA #IMPLIED
```

```
    Space CDATA #IMPLIED
```

```
    Type CDATA #IMPLIED
```

```
    COUNT CDATA #IMPLIED
```

```
ROWS CDATA #IMPLIED
```

```
COLS CDATA #IMPLIED
```

```
RVALUE CDATA #IMPLIED
```

```
CVALUE CDATA #IMPLIED
```

```
>
```



```
<WXML>
```

```
        // set backgroundcolor of report as pink and table color as lightgreen ; give the approximate fields
for reports
```

```
<REPORT TITLE='EMPLOYEE REPORT' Method='post' Url='http://localhost:8080/Programs/wstar/loginlist.wstar'
```

```
    color='pink' Tcolor='lightgreen' Name = 'JeminInformationTechnology (C) ltd' Space='yes' Type='submit' COUNT='1'
ROWS='1' COLS='12' RVALUE= 'EMPNAME, DEPT, DESIGNATION' CVALUE='3'/>
```

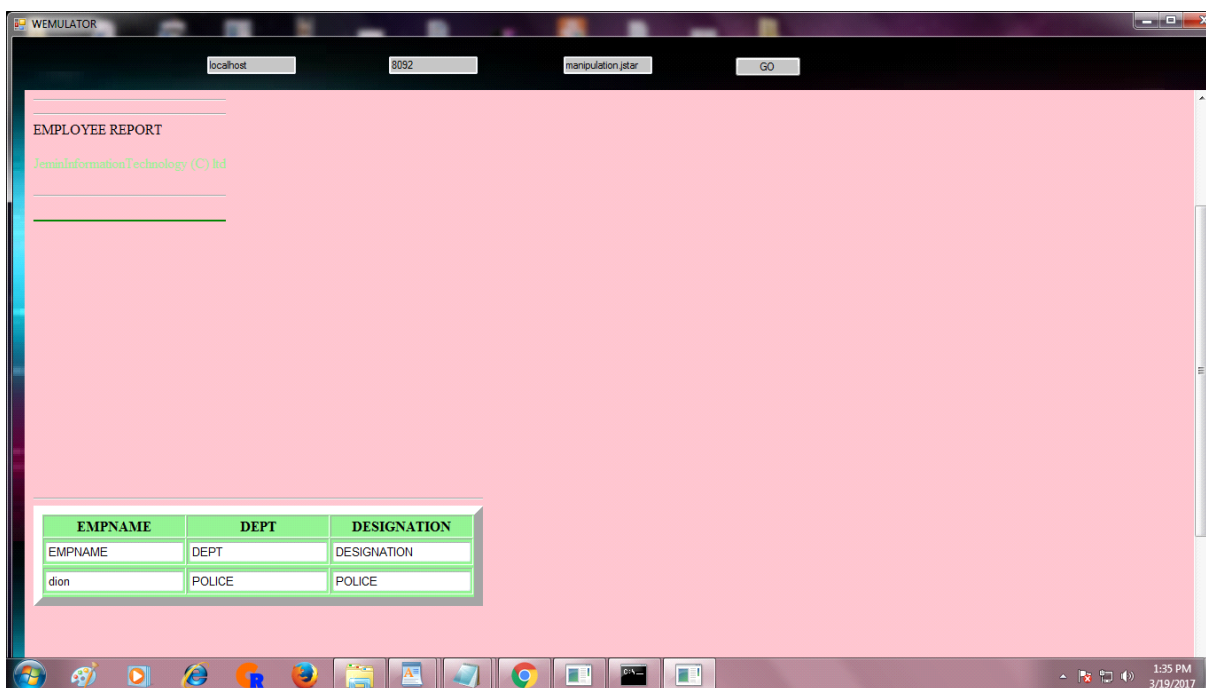
</WXML>

**Note:**

if WXML.wxml<PARSE>r("Report.wxml", 3,Secure.RetrieveSecure("output.wdba",0),"Report"); statement is added what happens?

it will parse the forms template and 3 indicates we are displaying data in report format

**Output in WEMULATOR:**




























**DIRECTORY STRUCTURE OF JSTAR****DIRECTORY STRUCTURE OF JSTAR**

**Note :** JStar has CDRUN.exe , Cdollar.exe, JStarserver.exe JSTAR-JAVA.exe ,CDollarv.4.exe,WLIMIXPROXY.exe , WNOSQLDBCONNECTOR.exe

are necessary files for JSTAR directory.

Jstar has JStar-cdollar,jstar-cdollarv.4.exe, jstar-cdc compiler,jstar-CDRun to run .wl files.



Name	Date modified	Type	Size
 Cdollar	7/8/2015 5:47 AM	Application	7 KB
 CDollarv.4	1/22/2016 4:54 PM	Application	8,014 KB
 CDRUN	8/10/2015 9:34 AM	Application	627 KB
 chart2	3/19/2017 10:09 AM	Application	4 KB
 CJAVA	3/10/2017 11:23 AM	Application	92 KB
 CR	8/10/2015 1:35 PM	Application	5,259 KB
 doc	4/23/2017 2:38 PM	Application	5 KB
 EMPLOYEE	1/15/2017 11:31 AM	Application	6 KB
 HierachialTree	9/1/2016 10:09 PM	Application	5 KB
 index	4/23/2017 2:52 PM	Application	4 KB
 J	9/12/2016 12:03 PM	Application	978 KB
 JAVA	3/20/2017 9:32 AM	Application	80 KB
 jgc	1/29/2005 10:35 PM	Application	86 KB
 JQuerytest	4/23/2017 2:56 PM	Application	5 KB
 JSTARSERVER	4/24/2017 8:02 AM	Application	2,079 KB
 manipulation	3/24/2017 6:03 PM	Application	6 KB
 objcopy	1/29/2005 10:35 PM	Application	558 KB
 Programs7	3/24/2017 5:24 PM	Application	5 KB
 Tree	4/18/2017 3:02 PM	Application	6 KB
 UtilityTimerExample	12/29/2016 10:13 ...	Application	4 KB
 WALLET1.0	7/8/2016 6:24 PM	Application	237 KB
 wdba	3/24/2017 4:42 PM	Application	5 KB
 WILMIXPROXY	12/28/2016 11:19 ...	Application	7 KB
 WNOSQLDBCONNECTOR	1/18/2017 7:46 PM	Application	5 KB
 wil	5/27/2015 10:11 AM	C File	1 KB

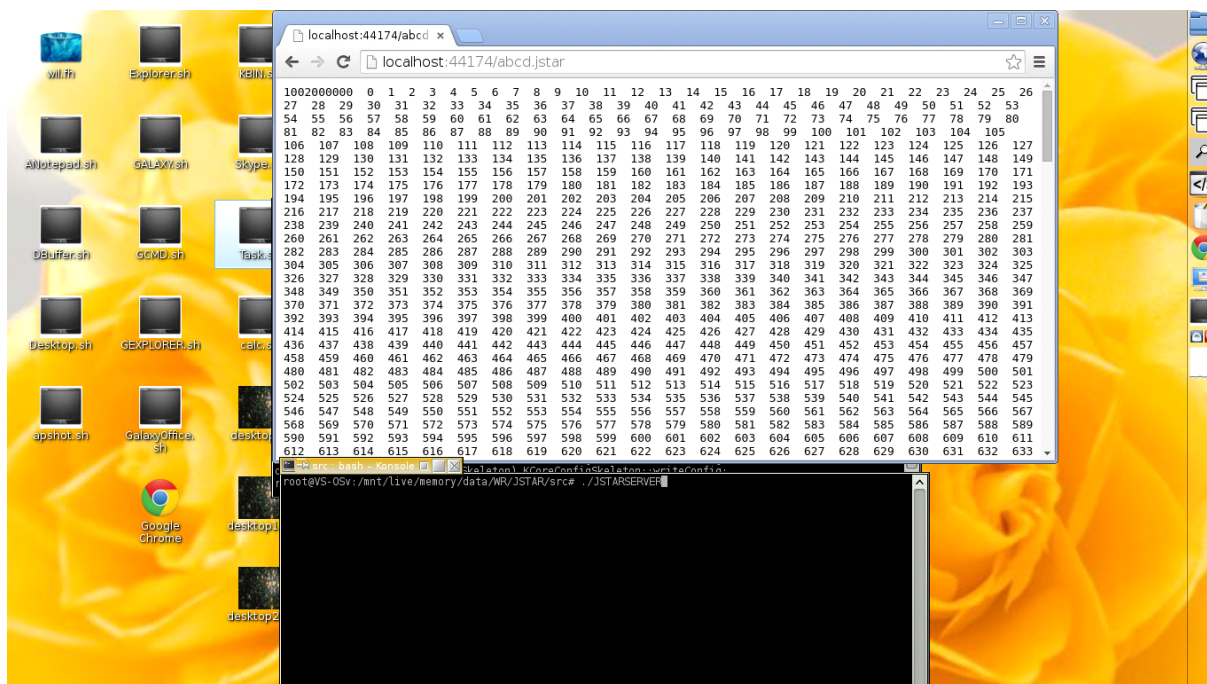
File version: 0.0.0.0  
Date created: 4/18/2017 4:55 PM  
Size: 627 KB

## UNIT -8

### JSTAR with Linux Platform

#### JSTARSERVER Screenshot running in linux type os?

JStarserver running in Linux Type os VS-OS which automatically generates some random no port as shown which is given below



**JSTAR with WNOSQL-PLSQL**

=====

JSTAR with WNOSQL - PLSQL is mostly focused

on using with SLING , LINUX, VS-OS platform.

kindly see this screen shot for more details...

Step -1: Write a WNOSQL-PLSQL Program named Programs.exe

=====

and type this url in browser set the path

as linux path as /mnt/live/memory/data/jstarfolder/src/

now we get the following output as follows.

Note: by using WNOSQL PLSQL JSTAR will execute Query in batch manner.

**JSTAR\_LINUX-EXAMPLE-1**

=====

Register.html

=====

```
<html>
```

```
<head>
```

```
<title> USER REGISTRATION</title>
```

```
</head>
```

```
<body class="fancy">
```

```
<form action="http://localhost:8092/Jquerytest.jstar" method="post" >
```

```
<div id="pageContainer">
```

```
<div id="pageContent">
```

```
<div id="chaptersAccordion">
```

```
<h2><a href="#chapter1">Enter your System Details</a></h2>
```

```
<div>
```

```
<p>Enter the Name: <input type="text" name="name" size="25" /></p>
```

```
<p>Enter your Username : <input type="text" name="uname" size="15" /></p>
```

```
<p>Enter the password : <input type="password" name="password" size="25" /></p>
```

```
</div>
```

```
<h2><a href="#chapter2">REGISTER</a></h2>
```

```
<div>
```

```
<input type="submit" name="Click">
```

```
<input type="reset" name="Clear">
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
index.jstar
```

```
<JSTAR>
```

```
<LOGIC>
```

```
public class index
```

```
{  
  
public void main( )  
{  
  
$HTML.displayhtml("Register.html"); // print the contents of Register.html  
  
}  
  
}
```

</LOGIC>

</JSTAR>

Jquerytest.jstar

=====

<JSTAR>

<USE> <J>.util.\*;

```
<USE> <J>.io.*;
```

```
<USE> CDollar.WDBA.*; // load wdba packages
```

```
<USE> Securitydb.*; //load Securitydb packages
```

```
<USE> java.sql.*;
```

```
<USE> jxl.Cell;
```

```
<USE> jxl.Sheet;
```

```
<USE> jxl.Workbook;
```

```
<USE> jxl.read.biff.Biff<EXE>;
```

```
<LOGIC>
```

```
public class JQuerytest
```

```
{
```

```
public void main( ) throws Biff<EXE>,IO<EXE>,<EXE>
```

```
{
```

```
<AList><String> arm1= <NEW> <AList><String>();
```

```
arm1.add("name");
```

```
arm1.add("uname");arm1.add("password");
```

```
// add all the form fields to arraylist
```

```
arm1.add("NOT");
```

```
<AList><String> armg=SPLITREQUEST.RESULT(arm1,"index.dsn",3,1);
```

```
// extract the contents of dsn file and print it in webpage
```

```
//dsn file contains parameters data , where as 3 indicates no of fields present in register form and 1 indicates the adjustment counter
```



```
// for selecting 3 fields
```

```
JSTAR.WriteLine("<table style='width:100%' bgcolor=gold>");
```

```
JSTAR.WriteLine("<tr>");
```

```
JSTAR.WriteLine(" <th>Name</th>");
```

```
JSTAR.WriteLine(" <th>Username</th>");
```

```
JSTAR.WriteLine(" <th>Password</th>");
```

```
JSTAR.WriteLine("</tr>");
```

```
JSTAR.WriteLine("<tr>");
```

```
for (int i=0;i<armg.size();i++)
```

```
{
```

```
JSTAR.WriteLine("<td>"+armg.get(i)+"</td>");
```

```
}

JSTAR.WriteLine("</tr>");

JSTAR.WriteLine("</table>");


char c= ' ';


// give path for wdbasql

String g=WDBAJDOLLAR.WDBAJDOLLARS("datastores","USEDATABASE",
"dbpwds","/mnt/live/memory/data/WilmixSoftware/WDBA/WDBAJ1");


// give username and password

String t= WDBAJDOLLAR.WDBAJDOLLARS("dbuser","dbpwds",1,"wilmix78", "wilmix78",1,5,g);


// Display emp6 table values

90
```

```
String query ="SELECTRVAL from emp6 3 to 33 , 1 to 1 ?= C By 1 1 : {0} : {0} :{1}";
```

```
WDBASQL.WDBASQLQUERY(query,t);
```

```
// calculate table size of emp6 by using size() in WDBASQL.WDBASQLQUERY
```

```
String q23="TABLESIZE() from emp6 3 to 33 , 1 to 1 ?= C By 1 1 : {0} : {0} :{1}";
```

```
WDBASQL.WDBASQLQUERY(q23,t);
```

```
String t1="";
```

```
t1=armg.get(0).StringConvert()+","+armg.get(1).StringConvert()+","+armg.get(2).StringConvert();
```

```
// insert the values form emp6 from emp6table size
```

```
String s12 ="INSERTINTO from emp6 3 to "+ WDBASQL.WDBASQLQUERY(q23,t).size()+" , 1 to 1 ?= A By 1 1 : {0} :  
{"+t1+"} : {0}";
```

```
WDBASQL.WDBASQLQUERY(s12,t);
```

```
// List emp6 table values
```

```
String query ="SELECTRVAL from emp6 3 to 33 , 1 to 1 ?= C By 1 1 : {0} : {0} :{1}";
```

```
JSTAR.WriteLine(""+WDBASQL.WDBASQLQUERY(query,t));
```

```
}
```

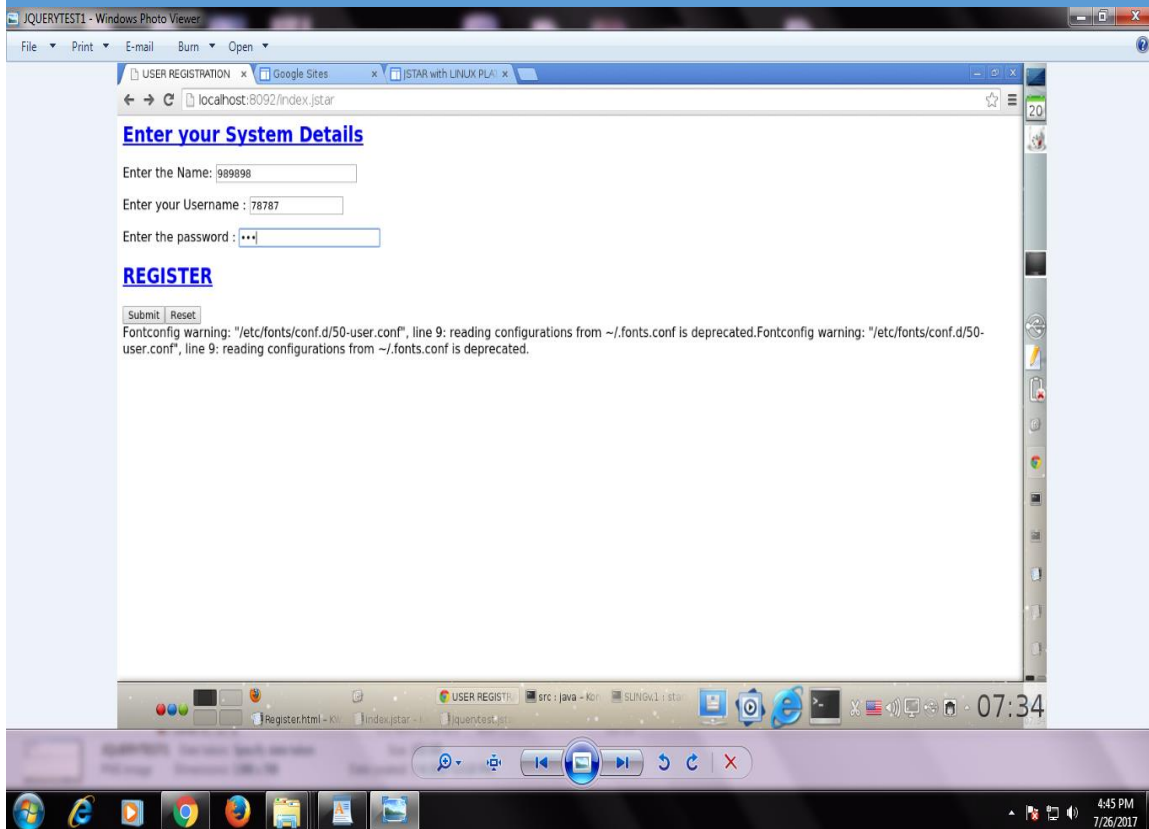
```
}
```

```
</LOGIC>
```

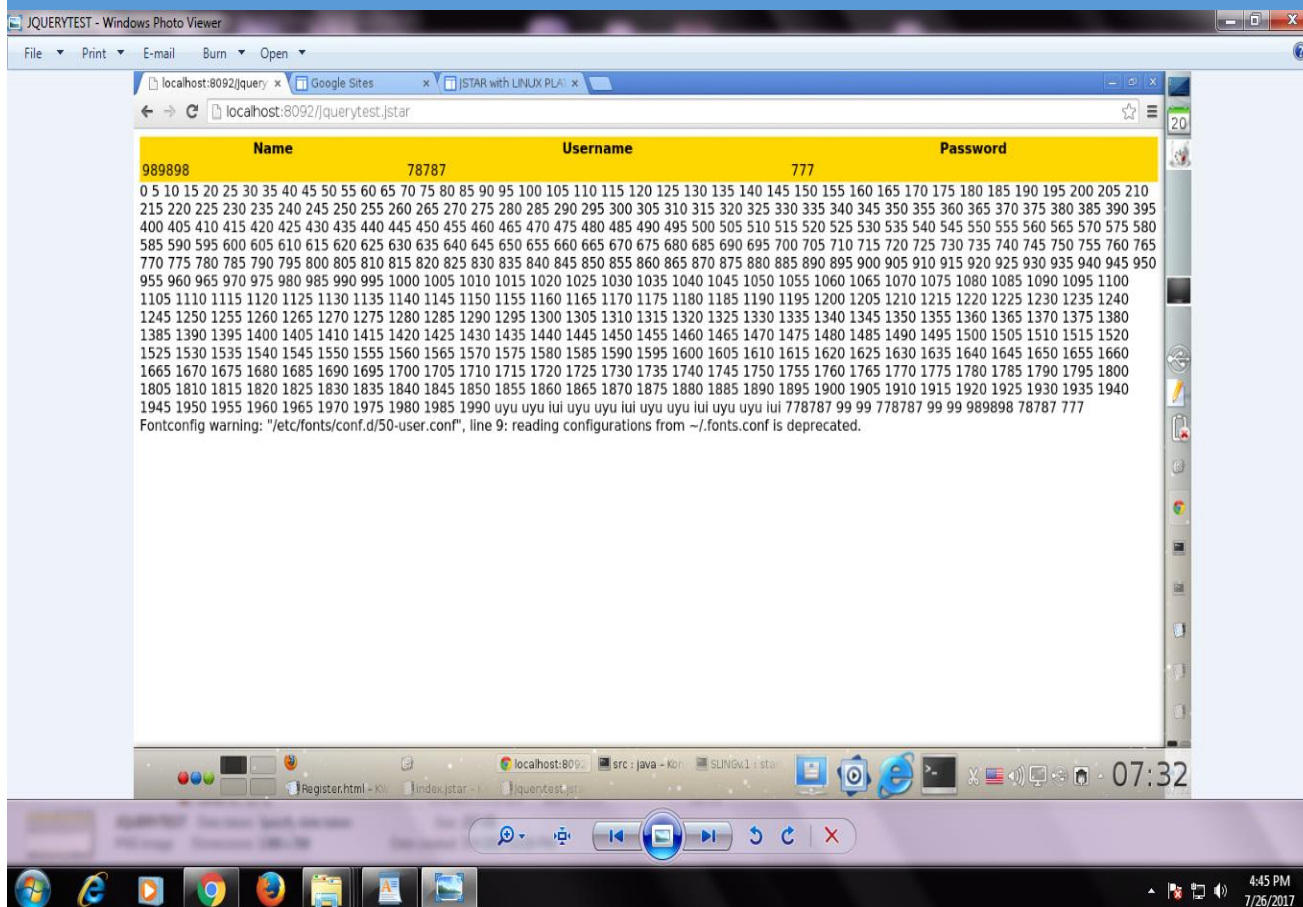
```
</JSTAR>
```

```
Input
```

```
===
```



### Output in SLINGv.1 os



=====

## JSTAR AJAX-EXAMPLE

**ARegister.html**

```
<html>

<head>

  <title> USER REGISTRATION</title>

</head>

<body class="fancy">
<form action="http://localhost:8092/AJAX.jstar" method="post" >

<div id="pageContainer">

  <div id="pageContent">

    <div id="chaptersAccordion">

      <h2><a href="#chapter1">Enter your System Details</a></h2>

      <div>

<p>Enter the Name: <input type="text" name="name" size="25" /></p>
<p>Enter your Username : <input type="text" name="uname" size="15" /></p>
<p>Enter the password : <input type="password" name="password" size="25" /></p>
```

```
</div>
```

```
<h2><a href="#chapter2">REGISTER</a></h2>
```

```
<div>
```

```
<input type="submit" name="Click">
```

```
<input type="reset" name="Clear">
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

### **AJAX.jstar**

```
<JSTAR>
```

```
<USE> <J>.util.*;
```

```
<USE> <J>.io.*;
```

```
<USE> CDollar.WDBA.*;
```

```
<USE> Securitydb.*;
```

```
<USE> java.sql.*;
```



```
<USE> jxl.Cell;
```

```
<USE> jxl.Sheet;
```

```
<USE> jxl.Workbook;
```

```
<USE> jxl.read.biff.Biff<EXE>;
```

```
<LOGIC>
```

```
public class AJAX
```

```
{
```

```
Shared int c=0;
```

```
public void main( ) throws Biff<EXE>,IO<EXE>,<EXE>
```

```
{
```

```
$HTML.displayhtml("ARegister.html");
```

```
//if (c==1)
```

```
{
```

```
<AList><String> arm1= <NEW> <AList><String>();
```

```
arm1.add("name");
```

```
arm1.add("uname");arm1.add("password");
```

```
arm1.add("NOT");
```

```
<AList><String> armg=SPLITREQUEST.RESULT(arm1,"AJAX.dsn",3,1);
```

```
// extract the contents of dsn file and print it in webpage
```

```
//dsn file contains parameters data , where as 10 indicates no of fields present in register form and 1 indicates the adjustment counter
```

```
// for selecting 10 fields
```

```
JSTAR.WriteLine("<table style='width:100%' bgcolor=gold>");
```

```
JSTAR.WriteLine("<tr>");
```

```
JSTAR.WriteLine(" <th>Name</th>");
```

```
JSTAR.WriteLine(" <th>Username</th>");
```

```
JSTAR.WriteLine(" <th>Password</th>");
```

```
JSTAR.WriteLine(" </tr>");
```

```
JSTAR.WriteLine(" <tr>");
```

```
for (int i=0;i<armg.size();i++)
```

```
{
```

```
JSTAR.WriteLine("<td>" + armg.get(i) + "</td>");
```

```
}
```

```
JSTAR.WriteLine(" </tr>");
```

```
JSTAR.WriteLine("</table>");
```

```
}
```

```
C++;
```

```
}
```

```
}
```

```
</LOGIC>
```

```
</JSTAR>
```

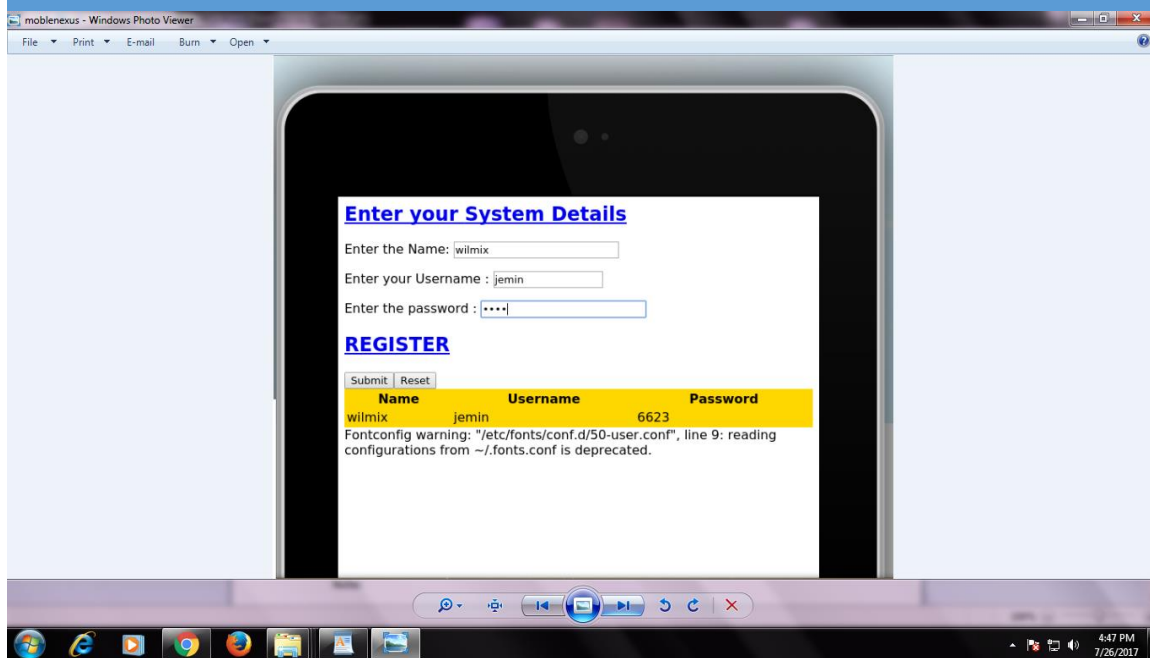
### **How to start JSTARSERVER in SLING or LINUX os?**

Enter the following in Konsole....

```
./JSTARSERVER231
```

.....Your JStarserver will start soon....

### **Output in Google Nexus Mobiles :**



### **Note:**

All the JSTAR Programs for Linux os for chapters like WILMIXTHEME, NEWSPAPER , CHARTS , etc are same when compared to Windows Os but here we use WDBAJdollar database with Jstar instead of WNOSQL with OAKJAVA7..

### **JSTAR Advanced Applications in webdesign**

=====

JSTAR CAN be used in most webdesign areas like Blogging ,Capatche,

AJAX,Site Search,Integrating Google maps, and involved in youtube publication.

=====

END OF LINUX/SLING OS with JSTAR Tutorial

=====

#### **JSTAR KEYWORDS similar to CDollarC keywords**

abstract boolean break byte

case <CATCH> char class const

continue default do double else

enum <--- final finally float

for goto if --> <USE>

instanceof int interface long native

<NEW> package private protected public

return short Shared strictfp <SUPER>

switch synchronized <IS>

throw throws transient <TRY> void volatile while

<% %>

#### OTHER KEYWORDS IN JSTAR

-----

AND -> AND operator

NOT -> NOT operator

# -> NOTEQUALS

RUN -> Runnable used in thread

TH-> Thread

<EXE> -> Exception

Friends -> Friend function

#### OTHER ATTRACTIVE SYMBOLS in JSTAR

-----

--> => implements

<-- => extends

#### ARRAYS

=====

ARRAY is to store a value in a location  
which uses stack dts..

SYNTAX for one dimension and multi dimension

=====

<DATATYPE> <variablename> Array [dimension1] ... [dimension-n]

eg)

```
int a11 Array [100];
```

```
    a11[0]=1000;
```

```
Jstar.WriteLine("/n"+a11[0]);
```

O/p

1000

## JSTAR Pointers

=====

What is Pointers?

Variables that hold memory address are called pointers.

Why we mainly use Pointers?

Pointers reduces the length and complexity of the program,

They increase the execution speed.

It holds the memory address..

SYNTAX of JSTAR Pointers:

-----

```
{*} <pointername> Pointers (<VALUE>);
```

## JSTAR Functions

=====

Functions are otherwise known as methods or apis can return or not return a value.

Functions are of two types with this cases they are

A) Function with or without return type using or without parameters.

b) JSTAR Operator overloading functions

SYNTAX:

=====

In JSTAR they are basically declared like this:

```
<visibility> <return type> <name>(<parameters>)
```

```
{
```

```
    <function code>
```

```
}
```



**JSTAR Operator Overloading function:**

=====

A function using operator to perform operations on a functions with parameters.

eg)

```
public Shared void operator *(int s1 ,int s2)
{
    s3=s1 * s2;
    Jstar.WriteLine(""+s3);
}
public Shared void LIB( )
{
    operator *(10,10);
    // You are passing * Multiply Operator in the main Program
    operator *(200,10000);
}
```

**Note:**

*If you know more about CDollarC fundamentals  
there is no need to STUDY JSTAR fundamentals.  
Since JSTAR follows CDollarC Programming Syntax.  
so kindly brush CDollarC fundamentals concepts  
before studying JStar webdesign Programming Concepts.*

**FAQS**

=====

**Why we use JSTAR for normal webdesign Programming Language?**

We use JSTAR for Remote Webpapplication WEBDESIGN .

**Why java did not interacts with JSTAR?**

Since java class file is different from CDollar class file.  
so JAVA7 (OAK JAVA) or JSTAR7 is completly focussed.

**Why JSTAR is Advanced than PHP? What are their differences?**

PHP source code is exposed and it is not converted to any bytecode file.

But JSTAR source code is not exposed and it is converted to bytecode file.

We can use OAKJAVA7 or JSTAR7 (obsucated) .exe file with JSTAR.

JStar Program also runs .exe file.

JSTAR also minimizes the code...

JSTAR is port independant....

JSTAR is focused on remotewebdesign but php is focused on webdesign.

JSTAR is used with WXML for displaying data in forms and report

**Why JSTAR is called a Terminal web design or remote webdesign?**

Normally when you compile the files in webserver and remoteserver publish in mobiles. but remoteserver displays Terminal Desktop where webserver cannot do....

**Will It work with CENTOS? How to install wine in CENTOS?**

kindly see this given url for more details...

and install it at CENTOS.

<https://billing.dotblock.com/knowledgebase/193/Install-Wine-on-CentOS-6-64bit.html>

Wine is most important since we use .dll and .exe for JSTAR Webdesign project.

don't run JSTAR program at hitache usb drive using centos.

**Why we use WEB1.0 with JSTAR? Is it a mandaltory one?**

Yes, it is important; Web1.0 is used for webservice , so we focus with JSTAR P.L.

**What is the Major Advantage of JSTAR programming Language?**

When JStar Server generates .class file for future use the filename.jstar will be emptied as blank filename.jstar file so the hackers cannot take the source code from the windows os through internet browsers..

So JSTAR prevents code stealing and provides good code security.

This is the major advantage of Jstar P.L over other Programming Language.where in Php p.L requires php code to be in filename.php and to be executed by WAMP server.

JStar server needs only .class or .exe or .html file..

So this is how JSTAR provides Good security for remote webdesign from hackers....

=====

## UNIT-9

# JSTAR ADVANCED CONCEPTS

=====

## JSTAR ADVANCED CONCEPTS

=====

JSTAR also follow wizard form controller like Spring framework do.

JSTAR follows to store all the incoming parameters in .dsn file and there is no need

for .xml file . Thus it simplifies the code. This is known as JSTAR Spring.

JSTAR also follows MVC design pattern...

JSTAR generates .wl files and thus it protect from hackers.

Now Let us see more about JSTAR wizard form controller with HTML and bootstrap template.

Design two forms one for entering father details and another for entering spouse details and print

those details use only jstar program and also use HTML with bootstrap template ...?

form.jstar

=====

<JSTAR>

<LOGIC>

public class form

{

public void main( )

{

\$HTML.displayhtml("formn.html");

}

}

</LOGIC>

</JSTAR>

form1.jstar

=====

<JSTAR>

<LOGIC>

```

public class form1

{

public void main( )
{

$HTML.displayhtml("UserForm1.html");

}

}

```

```

</LOGIC>
</JSTAR>

```

```

UserForm1.html
=====

```

```

<html>
<body>
<head>
<title>USERFORM</title>

</head>
<form name=f1 action="Jquerytest3.exe" method="post" >
<p>Your Name : <input type="text" name="name" /></p>
<p>Your Age : <input type="text" name="age" /></p>
<input type="submit" value="Add User" />
</form>

</body>
</html>

```

note: include file JQuerytest3.exe which is created using OAKJAVA7 program..

```
jquerytest3.web : OAKJAVA program
```

```
=====
```

```
<WEB>
```

```
<USE> <CDOLLARS>.util;
```

```
<PACK> JQuerytest36
```

```
{
```

```
    <CLASS> Prog
```

```
{
```

```
    public void main()
```

```
{
```

```
    ArrayList arm1 = new ArrayList();
```

```
    arm1.add("names");
```

```
    arm1.add("ages");
```

```
    arm1.add("name");
```

```
    arm1.add("age");
```

```
    arm1.add("NOT");
```

```

ArrayList armg= Request.Query(arm1,"Jquerytest3.dsn",4,1);

<PRINTLN>("<table style='width:100%' bgcolor=gold>");

<PRINTLN>("<tr>");

<PRINTLN>("  <th>Father NAME</th>");

<PRINTLN>("  <th>Age</th>");

<PRINTLN>("  <th>NAME</th>");

<PRINTLN>("  <th>Age</th>");

<PRINTLN>(" </tr>");

<PRINTLN>(" <tr>");

for (int i=0;i<armg.size()-1;i++)
{

<PRINTLN>("<td>"+armg.get(i).replace("&name","").replace("names","").replace("name","")+ "</td>");

}

<PRINTLN>(" </tr>");

```



```
<PRINTLN>("</table>");
```

```
    }
  }
}
```

## JSTAR SECURITY

```
=====
```

JSTAR SECURITY is achieved by OAKJAVA7.

Since OAKJAVA7 ".exe" prototype software is more obsucated.

OAKJAVA7 will execute more fast in the webbrowser.

OAKJAVA7 follows .dll to load the libraries..

Machine code is not easy for the hackers to trace...

In this way by using OAKJAVA7 with WNOSQL database

we attain JSTAR Security in windows /Linux platform.

## Advanced Features of JSTAR

```
=====
```

### A) SENDMAIL

```
=====
```

SendMail can be Achieved in jstar using Insert or Display API of APP().

But Sending gmail in Jstar Technology is a easy task but has some procedures to follow

```
<AList> arp = <NEW> <AList>();
```

```
arp.add(587); //add port
```

```
arp.add("smtp.gmail.com");// add smtp hostname whether it is yahoo or gmail
```

```
arp.add(TO@gmail.com); // mention sender mailid
```

```
arp.add("Form@gmail.com"); // mention receiptant mailid
```

```
arp.add("password"); // mention sender password
```

```
arp.add("About Your Lost / Found Item : "+id1); // add a Subject for the autogenerated mail
```

```
arp.add(" Congratulations NOT your "+id1+" is found.... This is a AutoGenerated mail for Searching the Item "); // add the matter of the //autogenerated mail
```

```
<NEW> APP().insert(arp,"email.properties"); // pass the arraylist in APP().insert API
```

```
// So what will this statement do ; it will insert into wdba format...
```

```
<NEW> APP().display(); // APP().display() will sent the mail automatically to you's friend or neighbour mailid..
```

```
// So it takes only few line to do this job....
```

### **Note:**

Make sure your gmail id permit this MAIL APPS...

follow google rules for sending autogenerated mail....

#### b) JSTAR Session Api

=====

It is used to maintain Sessions.

#### SYNTAX

=====

How to create a Session in JSTAR ?

```
SESSION u1 = <NEW> SESSION();
```

```
u1.AddSession(Stringindex,String); // it is used to Add a value or
string to the session index
```

How to get a value by using session index in JSTAR ?

```
JSTAR.WriteLine(""+SESSION.GetSession(Stringindex));
```

```
// it is used to get the value or string by using the session
index...
```

#### c) JSTAR ERROR API

=====

This Api is used for form validation purpose....

## SYNTAX

=====

```
ERROR u = new ERROR("ErrorFieldVALUE");
```

```
u.add(key,value);
```

```
JSTAR.WriteLine(""+u.l.get(value));
```

## JSTAR Integeration with Youtube

=====

JSTAR Program also allows your webpage to merge with you tube...

This is a new Advanced Feature in JSTAR...

## SYNTAX

=====

```
JSTARYOUTUBE.URL(int h, int w ,String url1,String smatter,String ematter)
```

h-> Height of the screen

w-> width of the screen

url1 -> Youtube Url for that topic

smatter -> Starting matter of the Topic

ematter -> Ending matter of the topic

JSTAR Advanced Applications in webdesign

=====

JSTAR CAN be used in most webdesign areas like Blogging ,Capatche, AJAX,Site Search,Integerating Google maps, and involved in youtube publication.

=====

### **JSTAR WILMIX PROXY SERVER**

#### **SYNTAX to Execute JSTAR PROXY SERVER**

JstarProxy <host> <remoteport> <localport>

remoteport => jstar remoteport

eg) WILMIXPROXY localhost 8090 80

now you can view the webpage in browser

using port 80...

This Proxy server is used for safe browsing...

=====

## UNIT-10:

=====

### JSTAR ADVANCED EXAMPLES

=====

#### Write a Newspaper Program using JSTAR?

Simillar to attractive designs with newspaper format is given below...

newspaper.html

=====

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet" href="http://localhost:8080/design10.WS">
```

```
    <title>TODAY NEWSPAPER</title>
```

```
</head>
```

```
<body>
```

```
<div class="box header">Indian-Asian Newspaper</div>
```

```
<div class="head">
```

```
    <div class="heading">
```

```
        <div class="saucer"><span style="font-style: italic;">At year 2020 , All
the world will focus JStar webdesign </span><br><span>JSTAR is more
attractive and easy to design</span></div>
```

```

    <header>Asian Newspaper</header>
</div>

    <div class="subhead"> BANGALORE - 25/6/2017 - One Page</div>
</div>
<div class="box content">

    <div class="box sidebar">

<div id="pageContainer">

    <div id="pageContent">

        <div id="chaptersAccordion">
<div class="content">
    <div class="collumns">
        <div class="collumn">

<figure class="figure">

        <figcaption class="box"><p align="center">This time is to focus
on Jstar webdesign..Welcome to solomon Technology</p></figcaption>
            </figure>

</div>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

Explanation: saucer will float the text

and <header> indicates newspaper header; boxheader will put the title in box.

subhead will display only subheading of the newspaper.figurecaption will arrange the important

news at left side...

Output using JSTAR Server for Modern most beautiful newspaper:





## JSTAR PAGINATION

=====

Write a Jstar program for nextAscendingSequence eg) 1,2 after you click next button

it will display 1,2,3,4 as output.. ?

NEXTASCSEQUENCE.jstar

=====

&lt;JSTAR&gt;

```
import <J>.util.*;
import <J>.sql.*;
import jxl.Cell;
import jxl.Sheet;
import jxl.Workbook;

import jxl.read.biff.Biff<EXE>;

import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.Write<EXE>;
```

&lt;LOGIC&gt;

```
public class NEXTASCSEQUENCE

{
```

```
Shared <AList><String> armg=null;
```

```
Shared int rowsPerPage=1;
```

```
Shared int rowcount=1;
```

```
public void main( ) throws Biff<EXE> ,<EXE>
{
```

```
int totalRows=0;
```

```
int r=0;
```

```
int c=0;
```

```
{
```

```
    <AList><String> arm1= <NEW> <AList><String>();
```

```
    arm1.add("id");
```

```
    arm1.add("page");
```

```
    arm1.add("!");
```

```
JSTAR.WriteLine("<form method=post action =  
http://localhost:8092/NEXTASCSEQUENCE.jstar >");
```

```
    armg=SPLITREQUEST.RESULT(arm1,"NEXTASCSEQUENCE.dsn",2,1);// where 2 indicates  
    2 fields
```

```
c=Integer.<PARSE>Int (armg.get(1).StringConvert());// retrieve updated data  
stored in .dsn file
```

```
r=Integer.<PARSE>Int (armg.get(0).StringConvert());// retrieve updated data  
stored in .dsn file
```

```
rowcount=c;
```

```
rowPerPage=r;
```

```
//update rowcount and rowpage...
```

```
}
```

```
PreparedStatement preparedStatement = null;
```

```
<TRY>
```

```
{
```

```
//step1 load the driver class
```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
//step2 create the connection object
```

```
Connection con=DriverManager.getConnection(
```

```
"jdbc:oracle:thin:@localhost:1521:xe","system","jemin");
```

```

{

preparedStatement =con.prepareStatement("SELECT itemname FROM FOUNDITEM  where
ROWNUM  between  "+rowsperPage+"

and  "+rowcount+" ORDER BY itemname ASC");

// this statement will list itemnames from 1 to 3 and 1 to 4 and
so-on where the itemname is sorted in Ascending order

}

ResultSet rs = preparedStatement.executeQuery();

// Create the table

    // Process the results. First dump out the column
    // headers as found in the ResultSetMetaData
    ResultSetMetaData rsmd = rs.getMetaData();

    int columnCount = rsmd.getColumnCount();

    // Start the table row

    // Now walk through the entire ResultSet and get each
    // row

int rows1=0;

JSTAR.WriteLine("<TABLE  bgcolor=yellow>");

JSTAR.WriteLine("<TH>FOUNDITEM</TH>");


    while (rs.next()) {
        rowcount++;
rows1++;

```

```
String id = rs.getString(1);

JSTAR.WriteLine("<Tr>");

JSTAR.WriteLine("<TD>" + id + "<td>");

JSTAR.WriteLine("</Tr>");

}

JSTAR.WriteLine("</table>");

JSTAR.WriteLine("<form method=post  action =  
http://localhost:8092/NEXTASCSEQUENCE.jstar >");

// take hidden form fields; and the rowcount incremented are stored in  
page field....

JSTAR.WriteLine("<input type=hidden name=id value="+rowsperPage+" />");
JSTAR.WriteLine("<input type=hidden name=page value="+rowcount+" />");

JSTAR.WriteLine("<input type=submit name=submit value=NextAscendingOrderSequences  
>");

// we note that id and page are fields and we consider as those  
field names

// and add to arm1

con.close();
```

```

}

<CATCH>(<EXE> e){ JSTAR.Writeln(e);}

}

}

</LOGIC>
</JSTAR>

```

```
=====
```

**JSTAR with GOOGLE CHARTS with STUDENT PASSING EXAMINATION RESULTS**

### **CHARTS.html**

```

<html>
  <head>
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>

```

```

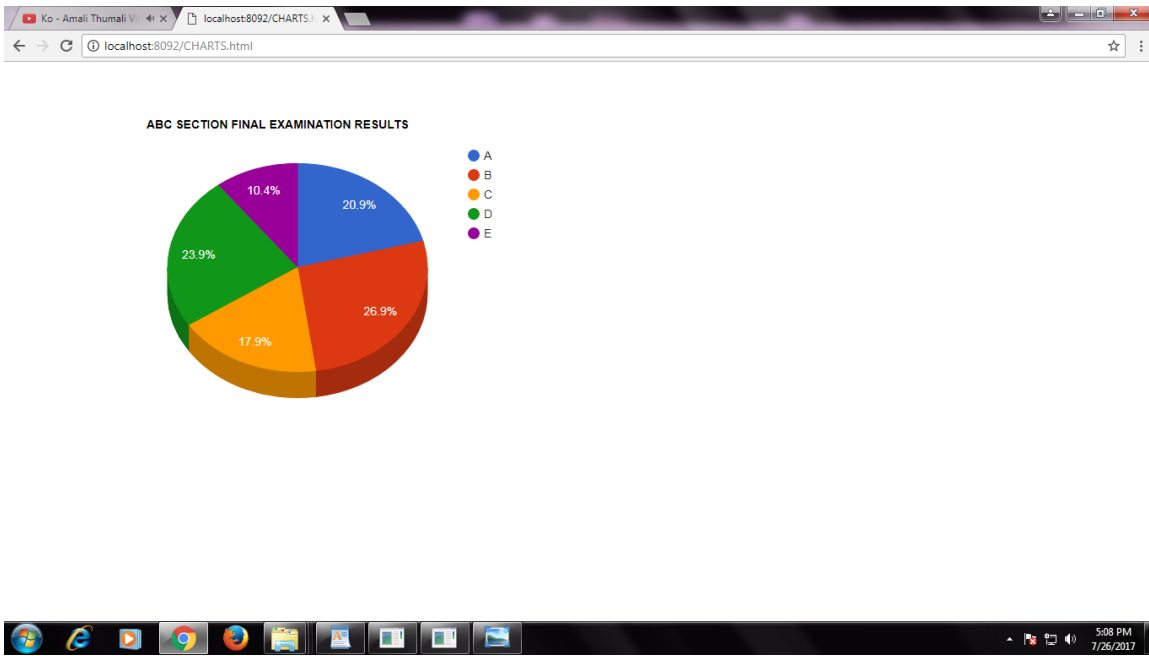
<script type="text/javascript">
  google.charts.load("current", {packages:["corechart"]});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['ABCSCHOOLSECTION', 'Percentage of students passed in public XII
examination'],
      ['A', 70],
      ['B', 90],
      ['C', 60],
      ['D', 80],
      ['E', 35]
    ]);

    var options = {
      title: 'ABC SECTION FINAL EXAMINATION RESULTS',
      is3D: true,
    };

    var chart = new
google.visualization.PieChart(document.getElementById('piechart_3d'));
    chart.draw(data, options);
  }
</script>
</head>
<body>
  <div id="piechart_3d" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

### **Output**



Note: If you want to learn more about charts, webgaming , animation ,etc you want to

go to Jdollarpart2(JWEB2.0) - WSTARAND WXML module...

=====



**JQUERY with JSTAR with Autocomplete feature**

=====

autocomplete1.html

=====

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Autocomplete - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-
ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
    $( function() {
      var availableTags =

```

autocomplete2.html

=====

```

;
    $( "#tags" ).autocomplete({
      source: availableTags
    });
  } );
</script>

```

```

</head>
<body>

<div class="ui-widget">
  <label for="tags">Tags: </label>
  <input id="tags">
</div>

</body>
</html>

```

Program12.web

=====

<WEB>

<PACK> Program12

```

{

  <CLASS> Prog
  {

    public void main()
    {

```

```

HTML.displayhtml("autocomplete1.html");

```

```

<PRINTLN>("[");
<PRINTLN>("'JWEB(JDollar)',");
<PRINTLN>("'DOTWEB',");
<PRINTLN>("'OAKJAVA7',");
<PRINTLN>("'PHP',");
<PRINTLN>("'PHYTHON',");

```

```
<PRINTLN> (" 'C', ");  
<PRINTLN> (" 'DOTNET', ");  
<PRINTLN> (" 'JAVA', ");  
<PRINTLN> (" 'GO', ");  
<PRINTLN> (" 'Angularjs', ");  
<PRINTLN> (" 'Android', ");  
<PRINTLN> (" 'Bootstrap.js', ");  
<PRINTLN> (" 'JavaScript'");  
<PRINTLN> ("]");
```

```
HTML.displayhtml ("autocomplete2.html");
```

```
    }  
  }  
}
```

Note:

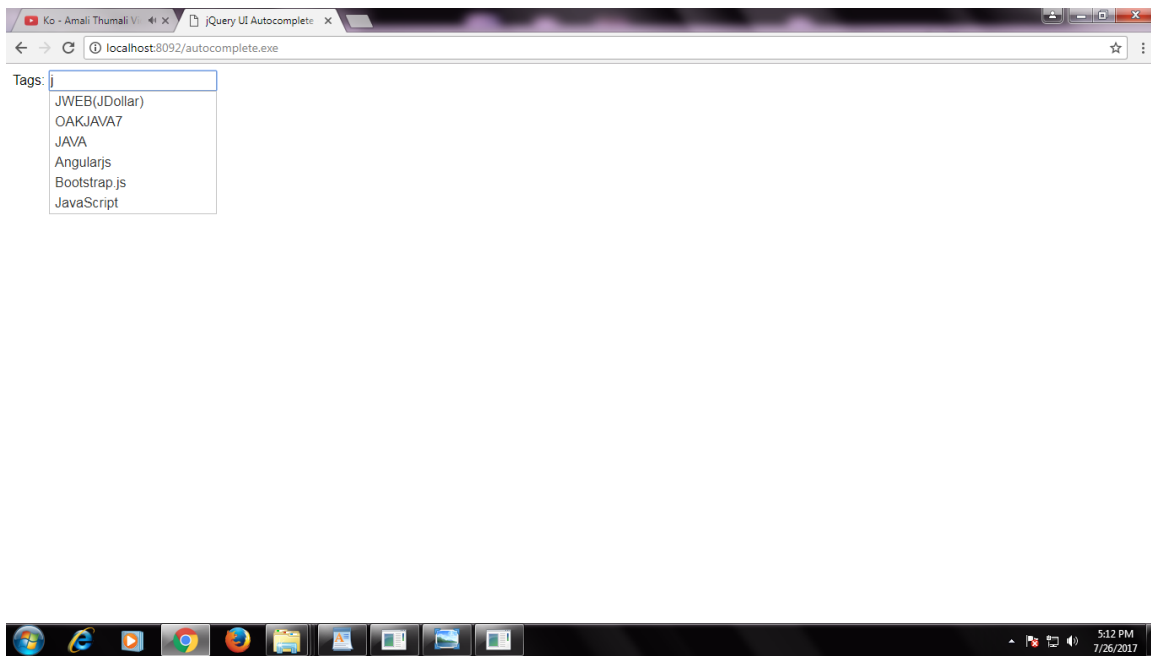
=====

Rename Program12.web.exe autocomplete.exe

and put it in Jstar server....

Output:

=====



---

## BLOG with JSTAR ProgrammingLanguage

---

### What is Blog?

A web page, typically one run by an individual or small group, that is written in an informal or conversational style.

So Syntax for datastructures used with JSTAR is given below

---

**BLOG SYNTAX**

=====

```
BLOG u = new BLOG("Blogheader");
```

```
u.add(String);
```

```
<PRINTLN> (" "+u.match(key,location));
```

```
u.CREATEBLOGPOST(key , blogtext);
```

```
<PRINTLN> (" "+u.rettl(location));
```

**Note:**

Use BLOG.dll in OAKJAVA7 properties file to

execute BLOG.exe created by OAKJAVA7...

=====

**Write a OAKJAVA7 or JSTAR7 program for that BLOG**

=====

```
{
```

```

BLOG u = new BLOG("Wilmix");

for (int i=0;i<=100;i++)
u.add("User"+i);

String s="HAI! Jstar is very interesting subject";

<PRINTLN>(""+u.match("User5",5));//used this statement for search blogpost
with Userid:User5

//Check User5 present at 5th location or not

u.CREATEBLOGPOST("0001" , s);

//Print only keyvalue ie)0001

<PRINTLN>(""+u.rettl(1));//used this statement for delete blogpost with id:
0001

u.SEARCHBLOGPOST("0001" , s);

//Print only Blogtextvalue ie) HAI! Jstar is very interesting subject

<PRINTLN>(""+u.rettl1(1));
// now you can store the blogtextvalue details at wnosql or oracle
database => (createblogpost)

//or print it in webpage =>(previewblogpost) or (EDITblogpost)

}

```

OUTPUT:

=====

true

0001

HAi! Jstar is very interesting subject

.....

=====

**JSTAR SessionTracking**

=====

Login.html

=====

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<script>
```

```
function validateForm() {
```

```
var uid= document.forms["my"]["userid"].value;
```

```
var pwd=document.forms["my"]["password"].value;
```

```

if (uid=="")

    alert("Kindly enter your Userid ");


if (pwd == "")
    alert("Kindly enter your Password");

}

</script>


// add wilmix theme


<meta name="viewport" content="width=device-width, initial-scale=1">
<title>jQuery UI Datepicker - Default functionality</title>
<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-
ui.css">
<link rel="stylesheet" href="/resources/demos/style.css">
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

</head>


<body bgcolor=gold class="fancy">
<form name="my" action="/Login2.jstar" method="post" onsubmit="return
validateForm();">


<div class="box header">User Login</div>

```



```

    <div class="box content">

        <div class="box sidebar">

<div id="pageContainer">

    <div id="pageContent">

        <div id="chaptersAccordion">

            <h2><a href="#chapter1">Enter your Login Details</a></h2>
            <div>

<p>Userid: <input type="text" name="userid" size="25" placeholder="Enter Your
Userid" /></p>

<p>Password: <input type="password" name="password" size="25" placeholder="Enter
Your Password" /></p>

<P> TYPE OF CUSTOMER : <select name="customertype">
<option selected>LOST
<option>FOUND
</select>

</p>

        <div>

    <div class="box content">
        <input type="submit" name="Click" onClick = "validateForm();">
<input type="reset" name="Clear">
        </div>

</div>

```

```
<div class="box footer"><h2><a
href="http://localhost:8092/Registerusers.html">Register users Click here to
continue...</a></h2></div>
```

```
<div class="box footer"> Login User(C) LTD</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
=====
```

```
Login2.jstar
```

```
=====
```

```
<JSTAR>
```

```
<USE> <J>.util.*;
```

```
<USE> <J>.sql.*;
```

```
<USE> jxl.Cell;
```

```
<USE> jxl.Sheet;
```

```
<USE> jxl.Workbook;
```

```
<USE> jxl.read.biff.Biff<EXE>;
```

```
<USE> jxl.write.Number;
```

```
<USE> jxl.write.WritableSheet;
```

```
<USE> jxl.write.WritableWorkbook;
```

```
<USE> jxl.write.Write<EXE>;
```

```
<LOGIC>
```

```
public class Login2
{

    public void main( ) throws Biff<EXE> ,<EXE>
    {

        String id=""; String pwd="";

        <AList><String> arm1= <NEW> <AList><String>();

        arm1.add("userid");
        arm1.add("password");

        arm1.add("customertype");
        arm1.add("!");

        <AList><String> armg=SPLITREQUEST.RESULT(arm1,".dsn",3,1);

        String insertTableSQL = "Select * from Register where userid = ? and password =
        ?";

        PreparedStatement preparedStatement = null;
```

```

<TRY>

{
//step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","jemin");

preparedStatement =con.prepareStatement(insertTableSQL);

preparedStatement.setString(1,armg.get(0).StringConvert());
preparedStatement.setString(2,armg.get(1).StringConvert());

ResultSet rs = preparedStatement.executeQuery();

String s= armg.get(2).StringConvert();

JSTAR.WriteLine("CUSTOMER="+s);

        while (rs.next()) {

                id = rs.getString("userid");
                pwd=rs.getString("password");

JSTAR.WriteLine("<body bgcolor=gold><form><p align=right> <b><font size=5
face=TimesNewRoman color=green>Hello </b> <b>" +id+" NOT
</font></b></p></form></body>" );

```

```
HTML.displayhtml("LOGOUT.html");

    }

    preparedStatement.executeUpdate();

//step5 close the connection object
con.close();

}

<CATCH>(<EXE> e) {

HTML.displayhtml("LOGOUT.html");

}

SESSION u1 = <NEW> SESSION();

u1.AddSession(""+id,id);// add userid to the session
```

```
if ((NOTpwd.<EQUALS>(armg.get(1).StringConvert())) AND
    (NOTid.<EQUALS>(armg.get(0).StringConvert()))))

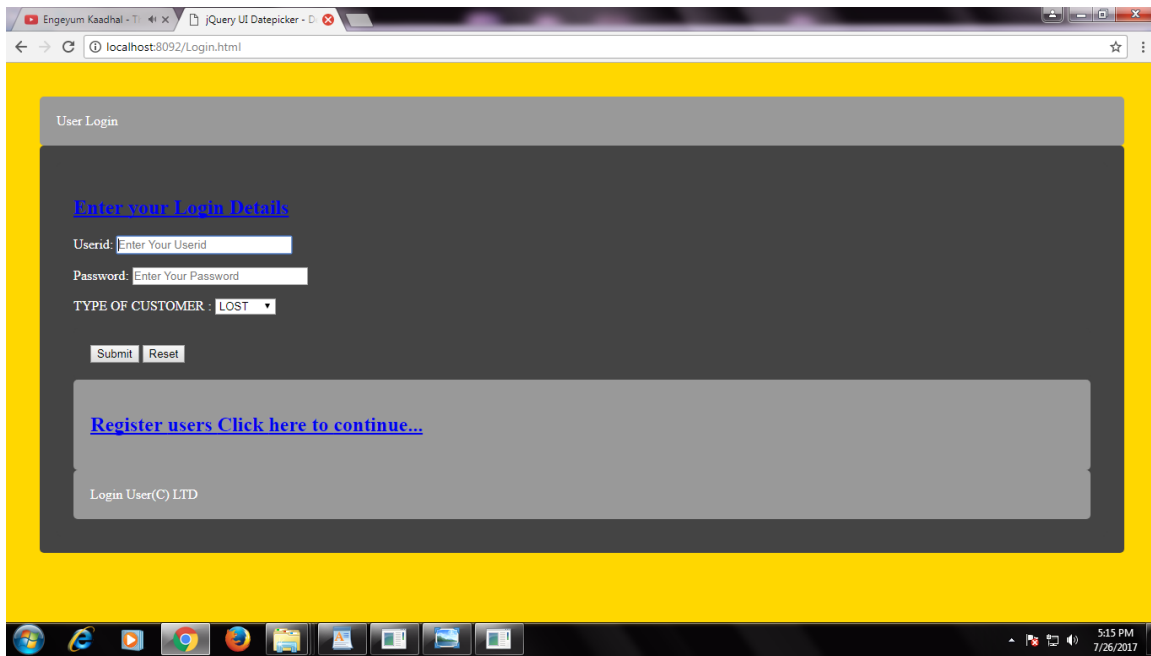
//if userid!=textboxuserid and password !=textpassword

HTML.displayhtml("LOGOUT1.html");// call LOGOUT

}

}

</LOGIC>
</JSTAR>
```

**Input: LoginPage-1**

=====

=====

**UNIT-11:**

=====

**JSTAR WITH PHP , OAKJAVA7 ,CDOLLAR,GDOLLAR, AND DOTNET**

=====

**JSTAR WITH PHP**

PHP fails in solving complex problems

for eg) AUTOGENERATION MAIL .

But JSTAR on the other hand solves complex problems

So JSTAR is used together with PHP in WEBHOSTING.

**JSTAR WITH DOTNET**

DOTNET behaves like C/C++ and C# behaves like JAVA.

But DOTNET is powerful in GUI design.

DOTNET is used in webdesign but doesn't

Have pure java oops and Advanced oops like Cdollar

And Gdollar . SO JSTAR is focused with DOTNET.



**JSTAR with OAKJAVA7**

OAKJAVA7 ".exe" works more fast and it has good security  
if some problems JSTAR , DOTNET ,PHP is not solved completely  
in that case JSTAR is used with OAKJAVA7 instead of JSTAR7.0.

**JSTAR with CDOLLAR and GDollar**

This is focused when a business requirement needs a powerful  
Datastructures to solve a problem or to minimize memory.

=====

UNIT-12:

=====

### **JSTAR MOCK EXERCISES**

=====

#### **JSTAR Programming EXERCISES:**

(1 \*100 =100 marks)

A) Write a JSTAR Program to design a BANK website ( 1\* 15 = 15 marks)

with transctions credit,debit,withdraw cash, discount ,etc.

[you can use java, C# ,etc]

B) Write a JSTAR Program for Electricity online Bill? (1\*5 = 5 marks)

c) write a JSTAR program to develop any company webpage.

(1\*10 = 10 marks)

d) Write a JSTAR Program to build a remotewebapplication to enter all

student details in a form and store it using wnosql database.

after that update it ,retrieve it and print the webpage. USE only PHP or DOTNET.

And use auto generate mail to sent the mark report/List to individual students of the class.

(1\*25=25 marks)

e) Write a JSTAR program with JQUERY or bootstrap to list the contents

from wnosql database and print it in table format. (1\*5= 5 marks)

f) Briefly explain JSTAR WORK FLOW. Write a JSTAR MVC program (1\*30 =30 marks)

to check whether the student name present or not from student form with wnosql  
and create fields using wnosql db with name,course,dateofjoin,dateoffinish, and status.  
and perform logic in model class

like

```
if (course=="java")
```

```
amt="2000"
```

```
else
```

```
if (course=="c/c++")
```

```
amt="10000"
```

```
else
```

```
if (course=="dotnet")
```

```
amt="15000"
```

```
else
```

```
if (course=="php")
```

```
amt="5000"
```

```
else
```

```
if (course=="mgt")
```

```
amt="25000"
```

if the student did not payed the fees mark the status as

"unpaid" otherwise mark the status "paid".

after that list all paid and unpaid people in

a seperate webpage.

g) Write briefly about the workflow of JSTAR Program.(1 \*5=5 marks)

h) EXPLAIN briefly with example RSSFEED and XML2JSON (1 \*5= 5 marks)

=====

## JSTAR Reference Documentation

=====

JSTAR Latest Tutorial will be updated in the JSTAR Tutorial Websites.....

=====

**Jemin Information Technology --- Copyright @ 2016 All Rights reserved**

=====