BaKaPlan - Baithne Ka Plan

1.0

Generated by Doxygen 1.8.5

Thu Dec 26 2013 02:02:51

Contents

Chapter 1

Main Page

This software creates seating plan for examinations. It provides five different strategies to create seat plan.

REQUIREMENTS:

```
1) GNU G++ Compiler
2) Configure public_html and cgi-bin in home
3) CGICC Library
4) MySQL Connector for C++
5) jwSMTP Library
6) exim4 mail server
7) Enable site(localhost/bakaplan) for BaKaPlan
```

Installation of requirements

1) GNU G++ Compiler
Run following command in terminal to install

```
$ sudo apt-get install g++
```

2) Configure public_html/cgi-bin folder for executing files on browser.
Assuming you already installed apache if not then run following command in terminal

```
$ sudo apt-get install apache2
```

Steps to configure public_html

```
$ mkdir ~/public_html

$ sudo a2enmod userdir

$ sudo service apache2 restart
```

Give 755 permissions to public_html directory

```
$ chmod -R 755 ~/public_html
```

Now open `http://localhost/~username` in browser. Here username is your login name.
Steps to configure cgi-bin in public_html

```
$ sudo a2enmod cgi

$ sudo a2enmod cgid

$ sudo service apache2 restart
$ cd ~/public_html
```

```
$ mkdir cgi-bin
```

```
$ cd /etc/apache2
```

```
$ sudo vim sites-available/default
```

Add following text in file:

```
ScriptAlias /cgi-bin/ /home/*/public_html/cgi-bin/
<Directory "/home/*/public_html/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    SetHandler cgi-script
    Order allow,deny
    Allow from all
</Directory>
```

Save it and then restart apache

```
$ sudo service apache2 restart
```

3) CGICC Library

Run following command in terminal

```
$ sudo apt-get install libcgicc-dev
```

OR

Download any latest package from `http://ftp.gnu.org/gnu/cgicc/`

Then run following commands in terminal

```
$ tar xzf cgicc-X.X.X.tar.gz
```

```
$ cd cgicc-X.X.X/
```

```
$ ./configure --prefix=/usr
```

```
$ make
```

```
$ sudo make install
```

4) MySQL and MySQL Connector for C++

Run following commands in terminal

```
$ sudo apt-get install mysql-server mysql-client
```

```
$ sudo apt-get install libmysql++
```

```
$ sudo apt-get install libmysql++-dev
```

5) jwSMTP Library

Download `jwSMTP` Library

Follow steps to install

```
$ cd ~/Downloads
$ tar -xzf jwsmtp-X.X.X.tar.gz
$ cd jwsmtp-X.X.X
$ ./configure
$ make
$ sudo make install
```

```
NOTE: If you got error in mailer.cpp or demo2.cpp file then
include string header file in that. And again run make
```

6) exim4 Mail Server

Run the following commands in terminal

```
$ sudo apt-get install exim4
$ sudo dpkg-reconfigure exim4-config
```

A Mail Server Configuration window will appear.

Follow the following instructions to configure the mail server.

```
1) The first page is just and introduction. Press ENTER
2) On the second page choose the second option i.e
   mail sent by smarthost; received via SMTP or fetchmail and
   press ENTER.
3) Next Keep the system mail name as it is and press ENTER.
4) Just Press Enter for the next page.
5) The next page asks you to enter IP addresses to listen on
   incoming SMTP connections. Leave it as it is and Press ENTER
6) Even on the next page let the value be the default one and
   Press ENTER.
7) Leave the next page as it is and Press ENTER.
8) The next page asks you for IP address or host name of the outgoing
   smarthost. Enter "smtp.example.com::587. Where example refers to
   gmail, yahoo or any other mail service provider and 587 is port number.
9) The Next page asks you if you want to hide local mail name in
   outgoing mail? Choose "No".
10)The Next asks you if you want to keep number of DNS-queries minimal?
   Choose "No".
11)On the next page choose the  delivery method for local mail as
   mbox format in /var/mail/.
12)Next page asks you if you want to split configuration into small
   files? Choose "No".
13)Next keep root and postmaster mail recipient empty.
```

Now terminal will show that MTA is being restarted.

After this is done, run the following command in terminal

```
$ sudo vim /etc/exim4/passwd.client
```

Add following in the file

```
*:USERNAME@example.com:PASSWORD.

Where, USERNAME is  a valid email address and PASSWORD is  password for USERNAME.
```

7) Enable site(bakaplan) using Apache

Do as following:

```
1) Open misc/bakaplan.conf file
2) Change username(*) with your username and save
3) Copy bakaplan.conf to /etc/apache2/sites-available
    $ sudo cp ~/public_html/cgi-bin/bakaplan/misc/bakpalan.conf
    /etc/apache2/site-available
4) Enable site
    $ sudo a2ensite bakaplan.conf
5) Reload apache
    $ sudo service apache2 reload
6) Now open site on browser
    localhost/bakaplan
```

INSTALLATION:

Check [INSTALLATION]() steps for using this software.

AUTHORS:

Mentor and Manager

Dr. Hardeep Singh Rai

Website: http://gndec.ac.in/~hsrai

Developers

Mandeep Kaur Inderpreet Singh Jaskaran Singh

Email: baithnekaplan [AT] gmail [DOT] com

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 name Namespace Reference

Variables

- std::map< string, string > **field**
- std::map< string, string > **table**

6.1.1 Detailed Description

Defining name map for declaring input field names used in html pages and column name of tables in database.

6.1.2 Variable Documentation

6.1.2.1 std::map<string, string> name::table

Initial value:

```
=
    {



        {"user", "User"},
        {"reg", "Registration"},
        {"project", "ProjectDetail"},
        {"session", "SessionDetail"},
        {"contact", "ContactDetail"},
        {"reset","ResetPassword"}
    }
```

Definition at line 106 of file fieldname.h.

Chapter 7

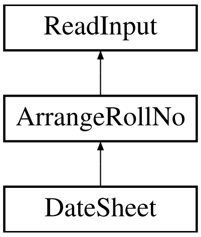
Class Documentation

7.1 ArrangeRollNo Class Reference

Arrange roll no class for sorting, removing redundant roll nos and also excludeing roll nos that are not for seating plan.

```
#include <arrangerollno.h>
```

Inheritance diagram for ArrangeRollNo:



Public Member Functions

- ArrangeRollNo ()

  Constructor.
- void SortRollNo (INT_VEC &rollno)

  Function for sorting roll no vector array.
- void RemoveRedundancy (INT_VEC &rollno)
- void RemoveNonEligibleRollNo (INT_VEC &rollno, INT_VEC &notIncludedRNo)

  Removing roll no that are not eligible.
- void AddPrefixWithRollNo (INT_VEC &rollno, STRING_VEC &prefix)

  Adding prefix with roll nos.
- void WriteArrangedRollNo ()

  Writing arranged roll nos to file as O/p file.
- void Main (string pID)

  Main function to call all functions.
- ∼ArrangeRollNo ()

  Destructor.

Protected Attributes

- INT_VEC rollNo
- INT_VEC notIncludedRNo
- INT_VEC **rollNoSize**
- INT_VEC **notIncludedRNoSize**
- STRING_VEC prefix
- STRING_2DVEC prefixRollNo
- ExpandRollNo expandRNo

7.1.1 Detailed Description

Arrange roll no class for sorting, removing redundant roll nos and also excludeing roll nos that are not for seating plan.

Include arrangerollno.h file

Include expandrollno.h file

Definition at line 30 of file arrangerollno.h.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 ArrangeRollNo::ArrangeRollNo ( )

Constructor.

Constructor

Definition at line 27 of file arrangerollno.cc.

7.1.2.2 ArrangeRollNo::∼ArrangeRollNo ( )

Destructor.

Desrtuctor

Definition at line 212 of file arrangerollno.cc.

7.1.3 Member Function Documentation

7.1.3.1 void ArrangeRollNo::AddPrefixWithRollNo (INT_VEC & *rollNo,* STRING_VEC & *prefix*)

Adding prefix with roll nos.

Adding prefix with roll nos

Parameters

rollNo	List of eligible roll nos
prefix	Prefix ie added with roll nos

Definition at line 107 of file arrangerollno.cc.

7.1.3.2 void ArrangeRollNo::Main (string *pID*)

Main function to call all functions.

Main method to call all function in order/sequence

Parameters

pID	Project ID

Definition at line 151 of file arrangerollno.cc.

7.1.3.3 void ArrangeRollNo::RemoveNonEligibleRollNo (INT_VEC & *rollNo,* INT_VEC & *notIncludedRNo*)

Removing roll no that are not eligible.

Removing non eligible roll nos

Parameters

rollNo	roll nos for seating plan
notIncludedRNo	Roll no list that is not eligible for seating plan

Definition at line 79 of file arrangerollno.cc.

7.1.3.4 void ArrangeRollNo::RemoveRedundancy (INT_VEC & *rollno*)

Removing redundancy from roll nos

Definition at line 51 of file arrangerollno.cc.

7.1.3.5 void ArrangeRollNo::SortRollNo (INT_VEC & *rollNo*)

Function for sorting roll no vector array.

Sorting Roll nos

Parameters

rollNo	For sorting array

Definition at line 39 of file arrangerollno.cc.

7.1.3.6 void ArrangeRollNo::WriteArrangedRollNo ( )

Writing arranged roll nos to file as O/p file.

Writing arranged roll no's into file

Definition at line 124 of file arrangerollno.cc.

7.1.4 Member Data Documentation

7.1.4.1 ExpandRollNo ArrangeRollNo::expandRNo [protected]

Object of ExpandRollNo class

Definition at line 45 of file arrangerollno.h.

7.1.4.2 INT_VEC ArrangeRollNo::notIncludedRNo [protected]

```
stroring rollnos
```

that are not included

Definition at line 34 of file arrangerollno.h.

7.1.4.3 STRING_VEC ArrangeRollNo::prefix [protected]

```
Storing prefix of
```

roll nos

Definition at line 40 of file arrangerollno.h.

7.1.4.4 STRING_2DVEC ArrangeRollNo::prefixRollNo [protected]

Roll no with prefix

Definition at line 43 of file arrangerollno.h.

7.1.4.5 INT_VEC ArrangeRollNo::rollNo [protected]

Reading roll nos

Definition at line 34 of file arrangerollno.h.

The documentation for this class was generated from the following files:

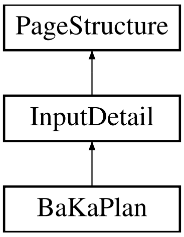
- src/cpp/backend/header/arrangerollno.h
- src/cpp/backend/arrangerollno.cc

7.2 BaKaPlan Class Reference

Main Project class.

`#include <bakaplan.h>`

Inheritance diagram for BaKaPlan:



Public Member Functions

- BaKaPlan ()

  constructor
- void Main (int step=0)

  Main Program.
- ∼BaKaPlan ()

  Desrtuctor.

Protected Attributes

- ProjectDetail **project**

Additional Inherited Members

7.2.1 Detailed Description

Main Project class.

Definition at line 24 of file bakaplan.h.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 BaKaPlan::∼BaKaPlan ( )

Desrtuctor.

Parameters

Definition at line 49 of file bakaplan.cc.

7.2.3 Member Function Documentation

7.2.3.1 void BaKaPlan::Main (int *step* = 0)

Main Program.

Parameters

step	Step to identify part of work flow

Definition at line 31 of file bakaplan.cc.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/bakaplan.h
- src/cpp/frontend/bakaplan.cc

7.3 Database Class Reference

Declaring functions for database comectivity and accessing database.

```
#include <database.h>
```

Public Member Functions

- Database ()

  Constructor to initialize database connectivity.
- void Query (string query)

  Function to execute any query.
- void UpdateQuery (string table, string column, string value, string where)

  Updating values in table.
- void SelectQuery (string query, STRING_VEC &result)

  Select Query for selecting single column from table.
- void InsertQuery (string query)

  Function for executing InsertQuery.
- void DeleteQuery (string table, string where)

Delete Query.

- void SelectSum (string column, string table, string projectID, string &result)
- void SelectColumn (STRING_VEC &result, string column, string table, string where="")
- void SelectProjectID (string &projectID)

  Select project ID from database.
- void InsertUserDetail (string emailID, string password)

  Insert User detail into User tabele.
- void InsertRegistrationDetail (string emailID, string registrationKey)

  Inserting Registration detail.
- void InsertResetPassword (string emailID, string resetKey)
- void InsertProjectDetail (string emailID, string projectName)

  Inserting Project Detail into database.
- void InsertClassDetail (string projectID, string className, string subjectName, string subjectCode)
- void InsertRollNoDetail (string projectID, string classID, string prefix, string startRNo, string endRNo, string notIncluded)
- void InsertRoomDetail (string projectID, string dateSheetID, string roomInfo, string centreName, string roomNo, string rows, string cols)
- void InsertStrategyDetail (string projectID, string dateSheetID, string strategyName)
- void InsertValidStrategy (string projectID, string dateID, string date, string selectedStrategy, string totalSeats, string totalStudents, string totalGroupSeats, string totalGroupStudents)
- void InsertDateSheet (string projectID, string date, string examCode)
- void InsertExamDetail (string projectID, string dateSheetID, string examName, string examSession, string examTime, string examVenue)
- void InsertSessionDetail (string emailID, string sessionKey)
- ∼Database ()

Protected Attributes

- MYSQL ∗ connect
- MYSQL_RES ∗ **res_set**
- MYSQL_ROW **row**
- unsigned int numrows
- STRING_VEC **vecTemp**
- string **i**
- string **j**
- string **query**

7.3.1 Detailed Description

Declaring functions for database comectivity and accessing database.

Definition at line 30 of file database.h.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 Database::Database ( )

Constructor to initialize database connectivity.

For making connection with database.

Definition at line 23 of file database.cc.

7.3.2.2 Database::∼Database ( )

For closing database connection

Class: Database Method: Database :: ∼Database()

Description: Destructor for closing database connectivity

Definition at line 351 of file database.cc.

7.3.3 Member Function Documentation

7.3.3.1 void Database::DeleteQuery (string *table*, string *where* = " ")

Delete Query.

Database Delete Query

Parameters

Definition at line 122 of file database.cc.

7.3.3.2 void Database::InsertClassDetail (string *projectID*, string *className*, string *subjectName*, string *subjectCode*)

Inserting class details

7.3.3.3 void Database::InsertDateSheet (string *projectID*, string *date*, string *examCode*)

Inserting datesheet details

7.3.3.4 void Database::InsertExamDetail (string *projectID*, string *dateSheetID*, string *examName*, string *examSession*, string *examTime*, string *examVenue*)

Inserting examimation details

7.3.3.5 void Database::InsertProjectDetail (string *emailID*, string *projectName*)

Inserting Project Detail into database.

Inserting project details

Definition at line 224 of file database.cc.

7.3.3.6 void Database::InsertQuery (string *query*)

Function for executing InsertQuery.

Database Insert Query

Parameters

query	query for inserting into value

Definition at line 109 of file database.cc.

7.3.3.7 void Database::InsertRegistrationDetail (string *emailID,* string *registrationKey*)

Inserting Registration detail.

Inserting into registration table

Parameters

emailID	emailID of user
registrationKey	Registration Key

Definition at line 199 of file database.cc.

7.3.3.8 void Database::InsertResetPassword (string *emailID,* string *resetKey*)

Insert into reset password table

Definition at line 208 of file database.cc.

7.3.3.9 void Database::InsertRollNoDetail (string *projectID,* string *classID,* string *prefix,* string *startRNo,* string *endRNo,* string *notIncluded*)

Inserting roll no details

7.3.3.10 void Database::InsertRoomDetail (string *projectID,* string *dateSheetID,* string *roomInfo,* string *centreName,* string *roomNo,* string *rows,* string *cols*)

Inserting room details

7.3.3.11 void Database::InsertSessionDetail (string *emailID,* string *sessionKey*)

Inserting into session detail table

Definition at line 325 of file database.cc.

7.3.3.12 void Database::InsertStrategyDetail (string *projectID,* string *dateSheetID,* string *strategyName*)

Inserting strategy details

7.3.3.13 void Database::InsertUserDetail (string *emailID,* string *password*)

Insert User detail into User tabele.

Inserting into User table

Parameters

emailID	Email Id
password	Password of user in encrypted from

Definition at line 182 of file database.cc.

7.3.3.14 void Database::InsertValidStrategy (string *projectID,* string *dateID,* string *date,* string *selectedStrategy,* string *totalSeats,* string *totalStudents,* string *totalGroupSeats,* string *totalGroupStudents*)

Inserting validation details for strategy

7.3.3.15 void Database::Query (string *query*)

Function to execute any query.

Database Query

Parameters

query	Holding MySQL Query

Definition at line 50 of file database.cc.

7.3.3.16 void Database::SelectColumn (STRING_VEC & *result,* string *column,* string *table,* string *where* = " ")

Select column from table

Definition at line 140 of file database.cc.

7.3.3.17 void Database::SelectProjectID (string & *projectID*)

Select project ID from database.

Select Project ID from table

Parameters

projectID	Unique project ID

Definition at line 164 of file database.cc.

7.3.3.18 Database::SelectQuery (string *query,* STRING_VEC & *result*)

Select Query for selecting single column from table.

Selecting column from table.

Database Select Query

Parameters

query	Select query
result	For storing selected column valued in result variable
result	
column	
table	
where	

Definition at line 87 of file database.cc.

7.3.3.19 void Database::SelectSum (string *column,* string *table,* string *projectID,* string & *result*)

Select sum for column in table

7.3.3.20 void Database::UpdateQuery (string *table,* string *column,* string *value,* string *where*)

Updating values in table.

Database Update Query

Parameters

table	Name of table in database
column	Column name in table
value	Value of column
where	Where clause if any

Definition at line 68 of file database.cc.

7.3.4 Member Data Documentation

7.3.4.1 MYSQL∗ Database::connect [protected]

MySQL connectivity Variables

Definition at line 34 of file database.h.

7.3.4.2 unsigned int Database::numrows [protected]

Unsigned int variable

Definition at line 38 of file database.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/database.h
- src/cpp/frontend/database.cc

7.4 DataSheet Class Reference

7.4.1 Detailed Description

Include datesheet.h file

The documentation for this class was generated from the following file:

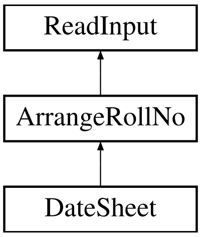
- src/cpp/backend/datesheet.cc

7.5 DateSheet Class Reference

DateSheet class for reading datesheet detail, class detail and create datesheet.out file for seating plan.

```
#include <datesheet.h>
```

Inheritance diagram for DateSheet:



Public Member Functions

- DateSheet ()
- void AddRollNoWithDateSheet ()

  Adding roll no to with exam subject code in datesheet.
- void WriteDateSheet (string projectID)
- void Main (string pID)
- ∼DateSheet ()

Additional Inherited Members

7.5.1 Detailed Description

DateSheet class for reading datesheet detail, class detail and create datesheet.out file for seating plan.

Include arrangerollno.h header file

Definition at line 27 of file datesheet.h.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 DateSheet::DateSheet ( )

Constructor

Definition at line 27 of file datesheet.cc.

7.5.2.2 DateSheet::∼DateSheet ( )

Desructor

Definition at line 161 of file datesheet.cc.

7.5.3 Member Function Documentation

7.5.3.1 void DateSheet::AddRollNoWithDateSheet ( )

Adding roll no to with exam subject code in datesheet.

Adding roll nos with date sheet

Definition at line 38 of file datesheet.cc.

7.5.3.2 void DateSheet::Main (string *pID*)

Main function to call other functions

Definition at line 143 of file datesheet.cc.

7.5.3.3 void DateSheet::WriteDateSheet (string *projectID*)

Writing datesheet.out file i.e. used for seating plan

Definition at line 99 of file datesheet.cc.

The documentation for this class was generated from the following files:

- src/cpp/backend/header/datesheet.h
- src/cpp/backend/datesheet.cc

7.6 ExapandRollNo Class Reference

Class for expanding roll nos.

```
#include <expandrollno.h>
```

7.6.1 Detailed Description

Class for expanding roll nos.

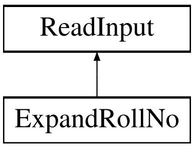
Include readinput.h

The documentation for this class was generated from the following file:

- src/cpp/backend/header/expandrollno.h

7.7 ExpandRollNo Class Reference

Inheritance diagram for ExpandRollNo:



Public Member Functions

- ExpandRollNo ()
- template<typename OutIter >
  bool ExpandRollNoList (istream &is, OutIter out)

  Templete for expanding roll nos.
- void ExpandRollNos (string projectID)

  Expanding Roll nos.
- void WriteExpandRollNo (string projectID)

  Write expanded roll nos into file.
- ∼ExpandRollNo ()

  Destructor.

Protected Attributes

- INT_VEC **rollNoSize**
- INT_VEC **notIncludedRNoSize**

7.7.1 Detailed Description

Definition at line 27 of file expandrollno.h.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 ExpandRollNo::ExpandRollNo ( )

Constructor

include expandrollno.h

Definition at line 22 of file expandrollno.cc.

7.7.2.2 ExpandRollNo::∼ExpandRollNo ( )

Destructor.

Destructor

Definition at line 195 of file expandrollno.cc.

7.7.3 Member Function Documentation

7.7.3.1 template<typename OutIter > bool ExpandRollNo::ExpandRollNoList (istream & *is*, OutIter *out*)

Templete for expanding roll nos.

Template for expanding roll no

Definition at line 94 of file expandrollno.cc.

7.7.3.2 void ExpandRollNo::ExpandRollNos (string *projectID*)

Expanding Roll nos.

For expanding roll no and calling ExpandRollNoList templete

Parameters

projectID	Project ID of user's project

Definition at line 34 of file expandrollno.cc.

7.7.3.3 void ExpandRollNo::WriteExpandRollNo (string *projectID*)

Write expanded roll nos into file.

Write expanded roll no's in file

Parameters

projectID	Project ID of user's project

Definition at line 149 of file expandrollno.cc.

The documentation for this class was generated from the following files:

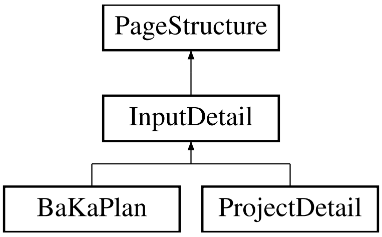
- src/cpp/backend/header/expandrollno.h
- src/cpp/backend/expandrollno.cc

7.8 InputDetail Class Reference

For declaring common variables and functions also used as base class.

```
#include <input_detail.h>
```

Inheritance diagram for InputDetail:



Public Member Functions

- InputDetail ()

  Constructor.
- string IntToString (int value)

  Converting integer to string.
- int StringToInt (string value)

  Converting String to Int.
- string FileName (string file, string projectID, int fileType)

  For creating filename w.r.t to projectID.
- ∼InputDetail ()

  Destructor.

Public Attributes

- Database db
- ReadInputField readField

Protected Attributes

- int **i**
- int **j**
- int **k**
- int l
- stringstream ss
- string **temp**

- string temp1
- string **projectID**
- string **emailID**
- string fileName
- string where
- string msg
- STRING_VEC vecTemp
- STRING_VEC oldProject
- STRING_VEC splitString
- ifstream inFile
- ofstream outFile
- string sessionID

7.8.1 Detailed Description

For declaring common variables and functions also used as base class.

Definition at line 29 of file input_detail.h.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 InputDetail::InputDetail ( )

Constructor.

< Shortname for namespace

Definition at line 23 of file input_detail.cc.

7.8.3 Member Function Documentation

7.8.3.1 string InputDetail::FileName (string *file,* string *projectID,* int *fileType*)

For creating filename w.r.t to projectID.

Parameters

projectID	Unique ID i.e. used to read file of user
file	I/O file Name
fileType	For defining file type i.e I/P or O/P file. If fileType == 1, Input file If fileType == 0(else), Output File

Returns

fileName This function will return file name with project ID

Definition at line 82 of file input_detail.cc.

7.8.3.2 string InputDetail::IntToString (int *value*)

Converting integer to string.

Parameters

value	Value to be converted into string

Definition at line 35 of file input_detail.cc.

7.8.3.3 int InputDetail::StringToInt (string *value*)

Converting String to Int.

Parameters

value	String value to be converted into int

Definition at line 47 of file input_detail.cc.

7.8.4 Member Data Documentation

7.8.4.1 Database InputDetail::db

DataBase class's object

Definition at line 60 of file input_detail.h.

7.8.4.2 string InputDetail::fileName [protected]

File name for opening file

Definition at line 42 of file input_detail.h.

7.8.4.3 ifstream InputDetail::inFile [protected]

For Reading file

Definition at line 54 of file input_detail.h.

7.8.4.4 int InputDetail::l [protected]

Looping Variables

Definition at line 34 of file input_detail.h.

7.8.4.5 string InputDetail::msg [protected]

For Displaying error message

Definition at line 42 of file input_detail.h.

7.8.4.6 STRING_VEC InputDetail::oldProject [protected]

For storing old projects if any

Definition at line 48 of file input_detail.h.

7.8.4.7 ofstream InputDetail::outFile [protected]

For writing file

Definition at line 55 of file input_detail.h.

7.8.4.8 ReadInputField InputDetail::readField

Reading Inpur fields

Definition at line 61 of file input_detail.h.

7.8.4.9 string InputDetail::sessionID [protected]

Session ID

Definition at line 56 of file input_detail.h.

7.8.4.10 STRING_VEC InputDetail::splitString [protected]

For storing values of splitted string

Definition at line 48 of file input_detail.h.

7.8.4.11 stringstream InputDetail::ss [protected]

For converting int to string

Definition at line 36 of file input_detail.h.

7.8.4.12 string InputDetail::temp1 [protected]

For temporary strorage

Definition at line 37 of file input_detail.h.

7.8.4.13 STRING_VEC InputDetail::vecTemp [protected]

string Vector temporary use

Definition at line 48 of file input_detail.h.

7.8.4.14 string InputDetail::where [protected]

Temp variable to store where clause

Definition at line 42 of file input_detail.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/input_detail.h
- src/cpp/frontend/input_detail.cc

7.9 jwsmtp::mailer Class Reference

Public Types

- enum {
  TO, **Cc**, **Bcc**, **SMTP_PORT** = 25,
  DNS_PORT = 53 }

- enum **authtype** { **LOGIN** = 1, **PLAIN** }

Public Member Functions

- **mailer** (const char ∗TOaddress, const char ∗FROMaddress, const char ∗Subject, const std::vector< char > &Message, const char ∗server="127.0.0.1", unsigned short Port=SMTP_PORT, bool MXLookup=true)
- **mailer** (const char ∗TOaddress, const char ∗FROMaddress, const char ∗Subject, const char ∗Message, const char ∗server="127.0.0.1", unsigned short Port=SMTP_PORT, bool MXLookup=true)
- **mailer** (bool MXLookup=false, unsigned short Port=SMTP_PORT)
- void **operator()** ()
- void **send** ()
- bool **attach** (const std::string &filename)
- bool **removeattachment** (const std::string &filename)
- bool **setmessage** (const std::string &newmessage)
- bool **setmessage** (const std::vector< char > &newmessage)
- bool **setmessageHTML** (const std::string &newmessage)
- bool **setmessageHTML** (const std::vector< char > &newmessage)
- bool **setmessageHTMLfile** (const std::string &filename)
- bool **setsubject** (const std::string &newSubject)
- bool **setserver** (const std::string &nameserver_or_smtpserver)
- bool **setsender** (const std::string &newsender)
- bool **addrecipient** (const std::string &newrecipient, short recipient_type=TO)
- bool **removerecipient** (const std::string &recipient)
- void **clearrecipients** ()
- void **clearattachments** ()
- void **reset** ()
- const std::string & **response** () const
- void **authtype** (const enum authtype Type)
- void **username** (const std::string &User)
- void **password** (const std::string &Pass)

Public Attributes

- enum jwsmtp::mailer::authtype **type**

Static Public Attributes

- static const enum
  jwsmtp::mailer:: { ... } **consts**

7.9.1 Detailed Description

Definition at line 33 of file mailer.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/jwsmtp/mailer.h
- src/cpp/frontend/header/jwsmtp/mailer.cpp

7.10 MD5 Class Reference

Public Types

- typedef unsigned int **size_type**

Public Member Functions

- **MD5** (const std::string &text)
- void **update** (const unsigned char ∗buf, size_type length)
- void **update** (const char ∗buf, size_type length)
- MD5 & **finalize** ()
- std::string **hexdigest** () const

Friends

- std::ostream & **operator**<< (std::ostream &, MD5 md5)

7.10.1 Detailed Description

Definition at line 50 of file md5.h.

The documentation for this class was generated from the following files:

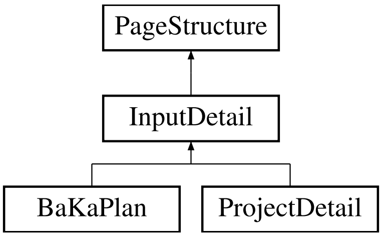
- src/cpp/frontend/header/md5.h
- src/cpp/frontend/md5.cc

7.11 PageStructure Class Reference

Declaring functions for common html tags and adding css and javascript files.

```
#include <page_structure.h>
```

Inheritance diagram for PageStructure:



Public Member Functions

- PageStructure ()

  Constructor.
- void CSS (string href)

  Adding CSS file in HTML Content.
- void JS (string src)

  Adding Javascript file in HTML Page.
- void CommonJSFiles ()

  Adding JS files for all pages.
- void CommonCSSFiles ()

  Adding css files for all pages.
- void SideMenu ()

  Side Menu Bar.
- ∼PageStructure ()

  Destructor.

7.11.1 Detailed Description

Declaring functions for common html tags and adding css and javascript files.

Definition at line 28 of file page_structure.h.

7.11.2 Member Function Documentation

7.11.2.1 void PageStructure::CSS (string *href*)

Adding CSS file in HTML Content.

Parameters

href	reference link

Definition at line 30 of file page_structure.cc.

7.11.2.2 void PageStructure::JS (string *src*)

Adding Javascript file in HTML Page.

Parameters

src	source link

Definition at line 42 of file page_structure.cc.

The documentation for this class was generated from the following files:

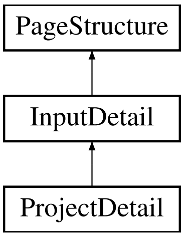
- src/cpp/frontend/header/page_structure.h
- src/cpp/frontend/page_structure.cc

7.12 ProjectDetail Class Reference

Reading Project details from database, browser and save values in database.

```
#include <project_detail.h>
```

Inheritance diagram for ProjectDetail:



Public Member Functions

- ProjectDetail ()

  Constructor.
- void ReadProjectDetail ()

  Reading Project detail filled by user.
- void WriteProjectDetail ()

Writing Project detail into file and database.

- void ProjectDetailPage ()

  HTML Project Detail Page.
- ∼ProjectDetail ()

  Desrtuctor.

Protected Attributes

- string **projectID**
- string **emailID**
- string **projectName**

Additional Inherited Members

7.12.1 Detailed Description

Reading Project details from database, browser and save values in database.

Definition at line 26 of file project_detail.h.

The documentation for this class was generated from the following files:

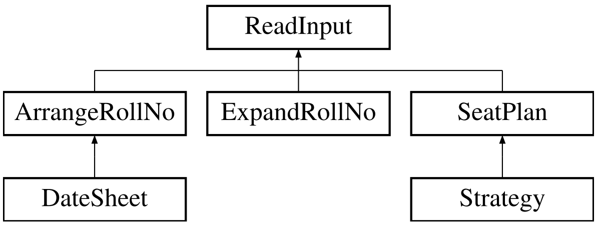
- src/cpp/frontend/header/project_detail.h
- src/cpp/frontend/project_detail.cc

7.13 ReadInput Class Reference

Class for reading input files ie reading roll no details, class detail, room detail, etc.

```
#include <readinput.h>
```

Inheritance diagram for ReadInput:



Public Member Functions

- ReadInput ()

  Constructor.
- string FileName (string file, string projectID, int fileType)

  For creating filename w.r.t to projectID.
- void ReadRollNoDetail (string projectID)

  Reading Roll No detail from file.
- void ReadClassDetail (string projectID)

  Reading class details.
- void ReadDateSheet (string projectID)

Reading datesheet.

- void ReadDateSheetOut (string projectID)

    Reading datesheet.out file.
- void ReadRoomDetail (string projectID)

    Reading room details.
- void ReadStrategy (string projectID)

    Function for reading strategy.
- void ReadExamDetail (string projectID)

    Reading Exam DEtails.
- ∼ReadInput ()

    Destructor.

Protected Attributes

- STRING_VEC prefix
- STRING_VEC rollNo
- STRING_VEC notIncluded
- STRING_VEC className
- STRING_VEC date
- STRING_VEC strategyOption
- STRING_VEC **examName**
- STRING_VEC **examSession**
- STRING_VEC **examTime**
- STRING_VEC **examVenue**
- STRING_VEC **examDate**
- STRING_2DVEC subjectCode
- STRING_2DVEC subjectName
- STRING_2DVEC examSubCode
- STRING_2DVEC examSubName
- STRING_2DVEC centreName
- STRING_3DVEC roomNo
- int totalClasses
- int totalDays
- INT_VEC totalSubjects
- INT_VEC totalExams
- INT_VEC totalCentres
- INT_2DVEC totalRooms
- INT_3DVEC rows
- INT_3DVEC cols
- ifstream inFile
- ofstream outFile
- string fileName
- string temp
- int **i**
- int **j**
- int **k**
- int **l**
- int **m**
- int n
- string projectID
- INT_2DVEC dateSheetRNoSize
- STRING_3DVEC dateSheetRNo

7.13.1 Detailed Description

Class for reading input files ie reading roll no details, class detail, room detail, etc.

Include local header file

Include readinput.h file

Definition at line 31 of file readinput.h.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 ReadInput::ReadInput ( )

Constructor.

Constructor

Definition at line 27 of file readinput.cc.

7.13.2.2 ReadInput::∼ReadInput ( )

Destructor.

Desrtuctor

Definition at line 378 of file readinput.cc.

7.13.3 Member Function Documentation

7.13.3.1 string ReadInput::FileName (string *file,* string *projectID,* int *fileType*)

For creating filename w.r.t to projectID.

Retruning filename w.r.t. to project ID

Parameters

projectID	Unique ID i.e. used to read file of user
file	I/O file Name
fileType	For defining file type i.e I/P or O/P file. If fileType == 1, Input file If fileType == 0(else), Output File

Returns

fileName This function will return file name with project ID

Definition at line 48 of file readinput.cc.

7.13.3.2 void ReadInput::ReadClassDetail (string *fileName*)

Reading class details.

Reading Class details from I/P file

Parameters

fileName	I/P filename for class detail

Definition at line 267 of file readinput.cc.

7.13.3.3 void ReadInput::ReadDateSheet (string *fileName*)

Reading datesheet.

Reading Datesheet

Parameters

fileName	Name of I/P file

Definition at line 312 of file readinput.cc.

7.13.3.4 void ReadInput::ReadDateSheetOut (string *projectID*)

Reading datesheet.out file.

Read date sheet with roll nos

Parameters

projectID	ProjectID

Definition at line 68 of file readinput.cc.

7.13.3.5 void ReadInput::ReadExamDetail (string *projectID*)

Reading Exam DEtails.

Reading exam details

Definition at line 232 of file readinput.cc.

7.13.3.6 void ReadInput::ReadRollNoDetail (string *fileName*)

Reading Roll No detail from file.

Reading Roll No details from I/P file

Parameters

projectID	Name of I/P file for roll nos.

Definition at line 201 of file readinput.cc.

7.13.3.7 void ReadInput::ReadRoomDetail (string *projectID*)

Reading room details.

Read Room Details

Parameters

projectID	projectId of user's project

Definition at line 134 of file readinput.cc.

7.13.3.8 void ReadInput::ReadStrategy (string *projectID*)

Function for reading strategy.

Reading strategy choice from strategy.in file

Parameters

projectID	projectId of user's project

Definition at line 355 of file readinput.cc.

7.13.4 Member Data Documentation

7.13.4.1 STRING_2DVEC ReadInput::centreName [protected]

Centre NAme

Definition at line 50 of file readinput.h.

7.13.4.2 STRING_VEC ReadInput::className [protected]

Branch/trade/class Name

Definition at line 35 of file readinput.h.

7.13.4.3 INT_3DVEC ReadInput::cols [protected]

columns in room

Definition at line 67 of file readinput.h.

7.13.4.4 STRING_VEC ReadInput::date [protected]

Date for examination

Definition at line 35 of file readinput.h.

7.13.4.5 STRING_3DVEC ReadInput::dateSheetRNo [protected]

```
Roll Nos w.r.t
```

datesheet

Definition at line 83 of file readinput.h.

7.13.4.6 INT_2DVEC ReadInput::dateSheetRNoSize [protected]

```
Total roll nos
```

for exam code

Definition at line 81 of file readinput.h.

7.13.4.7 STRING_2DVEC ReadInput::examSubCode [protected]

Examination/subject code

Definition at line 50 of file readinput.h.

7.13.4.8 STRING_2DVEC ReadInput::examSubName [protected]

Exam/ subejct Name

Definition at line 50 of file readinput.h.

7.13.4.9 string ReadInput::fileName [protected]

I/O file name

Definition at line 73 of file readinput.h.

7.13.4.10 ifstream ReadInput::inFile [protected]

For reading data from file

Definition at line 70 of file readinput.h.

7.13.4.11 int ReadInput::n [protected]

for looping

Definition at line 76 of file readinput.h.

7.13.4.12 STRING_VEC ReadInput::notIncluded [protected]

Rollno ie not for seating plan

Definition at line 35 of file readinput.h.

7.13.4.13 ofstream ReadInput::outFile [protected]

for writing data into file

Definition at line 71 of file readinput.h.

7.13.4.14 STRING_VEC ReadInput::prefix [protected]

Prefix of rollno

Definition at line 35 of file readinput.h.

7.13.4.15 string ReadInput::projectID [protected]

Project id of user’s project

Definition at line 78 of file readinput.h.

7.13.4.16 STRING_VEC ReadInput::rollNo [protected]

rollno range

Definition at line 35 of file readinput.h.

7.13.4.17 STRING_3DVEC ReadInput::roomNo [protected]

Room No

Definition at line 56 of file readinput.h.

7.13.4.18 INT_3DVEC ReadInput::rows [protected]

Rows in room

Definition at line 67 of file readinput.h.

7.13.4.19 STRING_VEC ReadInput::strategyOption [protected]

Strategy option selected by user

Definition at line 35 of file readinput.h.

7.13.4.20 STRING_2DVEC ReadInput::subjectCode [protected]

Subject Code

Definition at line 50 of file readinput.h.

7.13.4.21 STRING_2DVEC ReadInput::subjectName [protected]

Subject Name

Definition at line 50 of file readinput.h.

7.13.4.22 string ReadInput::temp [protected]

For temporary use

Definition at line 73 of file readinput.h.

7.13.4.23 INT_VEC ReadInput::totalCentres [protected]

Total centres

Definition at line 61 of file readinput.h.

7.13.4.24 int ReadInput::totalClasses [protected]

Total classes/branches/trades

Definition at line 58 of file readinput.h.

7.13.4.25 int ReadInput::totalDays [protected]

No. of days in datesheet

Definition at line 58 of file readinput.h.

7.13.4.26 INT_VEC ReadInput::totalExams [protected]

Total exams on 1 day

Definition at line 61 of file readinput.h.

7.13.4.27 INT_2DVEC ReadInput::totalRooms [protected]

total rooms in centre

Definition at line 65 of file readinput.h.

7.13.4.28 INT_VEC ReadInput::totalSubjects [protected]

Total subejcts of class

Definition at line 61 of file readinput.h.

The documentation for this class was generated from the following files:

- src/cpp/backend/header/readinput.h
- src/cpp/backend/readinput.cc

7.14 ReadInputField Class Reference

For reading values of fields like textbox, select, etc.

```
#include <read_inputfield.h>
```

Public Member Functions

- string ReadFieldValue (string fieldName)

  Read field's value and return it as string.
- string ReadFieldValue (string fieldName, int fieldNo)

  Read field's value and return it as string.
- void SetCookie (string cookieName, string value)

  Set session cookie.

Protected Attributes

- string fieldValue
- Cgicc formData
- form_iterator fi

7.14.1 Detailed Description

For reading values of fields like textbox, select, etc.

Definition at line 29 of file read_inputfield.h.

7.14.2 Member Function Documentation

7.14.2.1 string ReadInputField::ReadFieldValue (string *fieldName*)

Read field's value and return it as string.

Parameters

fieldName	Input field name

Definition at line 22 of file read_inputfield.cc.

7.14.2.2 string ReadInputField::ReadFieldValue (string *fieldName,* int *fieldNo*)

Read field's value and return it as string.

Parameters

fieldName	Input field name
fieldNo	Input field no to identify fields with same name.

Definition at line 44 of file read_inputfield.cc.

7.14.2.3 void ReadInputField::SetCookie (string *cookieName,* string *value*)

Set session cookie.

Parameters

cookieName	Name of cookie
value	Value of cookie

Definition at line 60 of file read_inputfield.cc.

7.14.3 Member Data Documentation

7.14.3.1 form_iterator ReadInputField::fi [protected]

Form Iterator

Definition at line 36 of file read_inputfield.h.

7.14.3.2 string ReadInputField::fieldValue [protected]

Input Field Value

Definition at line 32 of file read_inputfield.h.

7.14.3.3 Cgicc ReadInputField::formData [protected]

Read Form Data

Definition at line 35 of file read_inputfield.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/read_inputfield.h
- src/cpp/frontend/read_inputfield.cc

7.15 Seatplan Class Reference

The documentation for this class was generated from the following file:

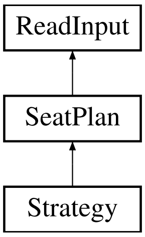
- src/cpp/backend/seatplan.cc

7.16 SeatPlan Class Reference

SeatPlan CLass for generating seating plan.

```
#include <report.h>
```

Inheritance diagram for SeatPlan:



Public Member Functions

- RoomReport ()
- void ReadExamDetail (string projectID)
- void ReadSeatPlan (string projectID)
- void WriteHTMLFile (string projectID)
- ∼RoomReport ()
- SeatPlan ()

  Constructor.
- void **SetRollNo** (int strategy, int i)
- string **RollNo** (int s)
- void **SeatingPlan** (int strategy, int i)
- void **CreateFile** (string projectID)
- void **WriteSeatPlan** (string projectID, int i)
- void WriteHTMLFile (string projectID, int i)

  Creating HTML file.
- void **WritePDFFile** (string projectID, int i)
- void **AddRollNoInfo** (string projectID, int i)
- ∼SeatPlan ()

  DEstructor.

Protected Attributes

- int **totalSeats**
- int **totalStudents**
- int **totalGroupSeats**
- int **day**
- INT_VEC **iTemp**
- INT_VEC **indexValue**
- INT_VEC **groupStudentSize**
- INT_VEC **seatSize**
- INT_VEC **size**
- INT_3DVEC **roomSize**
- STRING_VEC **subSubCode**
- STRING_2DVEC **seatRollNo**
- STRING_2DVEC **subRollNo**
- STRING_4DVEC **seat**

- int **textWidth**
- int **textWidth1**
- int **rectWidth**
- int **x**
- int **y**
- int **width**
- int **height**
- int **centre**
- int **room**
- int **room1**
- int **row**
- int **col**
- int **s**
- int **start**
- int **end**
- int **index**

7.16.1 Detailed Description

SeatPlan CLass for generating seating plan.

Include local header file

include readinput.h

include seatplan.h file

Definition at line 25 of file report.h.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 SeatPlan::∼RoomReport ( )

Destructor

7.16.2.2 SeatPlan::SeatPlan ( )

Constructor.

Constructor

Definition at line 28 of file seatplan.cc.

7.16.2.3 SeatPlan::∼SeatPlan ( )

DEstructor.

Destructor

Definition at line 593 of file seatplan.cc.

7.16.3 Member Function Documentation

7.16.3.1 void SeatPlan::ReadExamDetail (string *projectID*)

Read Exam Detail

7.16.3.2 void SeatPlan::ReadSeatPlan (string *projectID*)

Read Seat Plan

7.16.3.3 SeatPlan::RoomReport ( )

Constructor

7.16.3.4 void SeatPlan::WriteHTMLFile (string *projectID*)

WriteHTMLFile

7.16.3.5 void SeatPlan::WriteHTMLFile (string *projectID,* int *i*)

Creating HTML file.

Parameters

projectID	Project Id of seating plan project
i	For creating file accord to datesheet

Definition at line 311 of file seatplan.cc.

The documentation for this class was generated from the following files:

- src/cpp/backend/header/report.h
- src/cpp/backend/header/seatplan.h
- src/cpp/backend/seatplan.cc

7.17 SendMail Class Reference

Class for sending mail to user for registeration and for changing password.

```
#include <sendmail.h>
```

Public Member Functions

- SendMail ()

  Constructor.
- void SetMailData ()

  Settinf variable that are used to send mail.
- void SetHTMLMessage (string regKey, string mail)

  Setting body of mail(message ie send to user)
- void RegistrationMail (string setRecipient, string regKey)

  Sending registration mail to user.
- void ResetPasswordMail (string setRecipient, string regKey)

  Sending mail to user for changing password.
- void ContactMail (string setSender, string msg)

  Sending mail to developer from user.
- ∼SendMail ()

  Destructor.

Protected Attributes

- string setSender
- string setSubject
- string setMessage
- string setServer
- string htmlMessage
- string url

7.17.1 Detailed Description

Class for sending mail to user for registeration and for changing password.

Definition at line 34 of file sendmail.h.

7.17.2 Member Function Documentation

7.17.2.1 void SendMail::ContactMail (string *setSender,* string *msg*)

Sending mail to developer from user.

Contacl mail to developer send by user

Parameters

setSender	Sender's email id.
msg	Message send by user for developer.

Definition at line 144 of file sendmail.cc.

7.17.2.2 void SendMail::RegistrationMail (string *setRecipient,* string *regKey*)

Sending registration mail to user.

Registration emial for new user

Parameters

setRecipient	Reciever's email id
regKey	Registration key

Definition at line 89 of file sendmail.cc.

7.17.2.3 void SendMail::ResetPasswordMail (string *setRecipient,* string *regKey*)

Sending mail to user for changing password.

Sending mail to reset password

Parameters

setRecipient	Reciever's email id
regKey	Registration key

Definition at line 117 of file sendmail.cc.

7.17.2.4 void SendMail::SetHTMLMessage (string *regKey,* string *mail*)

Setting body of mail(message ie send to user)

HTML Message of mail

Parameters

regKey	Registration Key
mail	Mail type - regiter/confirm/reset-password

Definition at line 47 of file sendmail.cc.

7.17.2.5 void SendMail::SetMailData ( )

Settinf variable that are used to send mail.

Setting mail information

Definition at line 33 of file sendmail.cc.

7.17.3 Member Data Documentation

7.17.3.1 string SendMail::htmlMessage [protected]

Message in html form

Definition at line 38 of file sendmail.h.

7.17.3.2 string SendMail::setMessage [protected]

Plain/Rich text mail

Definition at line 38 of file sendmail.h.

7.17.3.3 string SendMail::setSender [protected]

Sender's email id

Definition at line 38 of file sendmail.h.

7.17.3.4 string SendMail::setServer [protected]

Server: localhost or domain name

Definition at line 38 of file sendmail.h.

7.17.3.5 string SendMail::setSubject [protected]

Subject of mail

Definition at line 38 of file sendmail.h.

7.17.3.6 string SendMail::url [protected]

Confirmation link or url

Definition at line 38 of file sendmail.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/sendmail.h
- src/cpp/frontend/sendmail.cc

7.18 jwsmtp::SOCKADDR_IN Struct Reference

Public Member Functions

- **SOCKADDR_IN** (const std::string &address, unsigned short port, short family=AF_INET)
- **SOCKADDR_IN** (const SOCKADDR_IN &addr)
- SOCKADDR_IN **operator=** (const SOCKADDR_IN &addr)
- **operator bool** () const
- **operator const sockaddr_in** () const
- **operator const sockaddr** () const
- size_t **get_size** () const
- char ∗ **get_sin_addr** ()
- void **set_port** (unsigned short newport)
- void **set_ip** (const std::string &newip)
- void **zeroaddress** ()

Public Attributes

- sockaddr_in **ADDR**

7.18.1 Detailed Description

Definition at line 84 of file compat.h.

The documentation for this struct was generated from the following file:

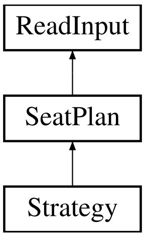
- src/cpp/frontend/header/jwsmtp/compat.h

7.19 Strategy Class Reference

Strategy class to check selected strategy is valid or not for seating plan.

```
#include <strategy.h>
```

Inheritance diagram for Strategy:



Public Member Functions

- Strategy ()

  Constructor.
- void ChooseStrategy ()

  Select Strategy.
- void **TotalSeats** (int strategy, int i)
- void **TotalStudents** (int i)

- void **CheckValidation** (int strategy, int i)
- void **TotalGroupStudents** (int strategy, int i)
- void **GroupCondition** (int strategy, int i)
- void Main (string pID)

  Main Function to call all other functions.
- string **IntToString** (int value)
- ∼Strategy ()

  Destructor.

Protected Attributes

- STRING_VEC strategyName
- int totalStrategy
- Database **db**
- stringstream **ss**

7.19.1 Detailed Description

Strategy class to check selected strategy is valid or not for seating plan.

include seatplan.h file

include strategy.h

Definition at line 30 of file strategy.h.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 Strategy::Strategy ( )

Constructor.

Constructor

Definition at line 28 of file strategy.cc.

7.19.2.2 Strategy::∼Strategy ( )

Destructor.

Destructor

Definition at line 301 of file strategy.cc.

7.19.3 Member Function Documentation

7.19.3.1 void Strategy::Main (string *pID*)

Main Function to call all other functions.

Parameters

pID	project ID of user's project

Definition at line 275 of file strategy.cc.

7.19.4 Member Data Documentation

7.19.4.1 STRING_VEC Strategy::strategyName `[protected]`

Name of Strategy

Definition at line 33 of file strategy.h.

7.19.4.2 int Strategy::totalStrategy `[protected]`

Total Strategies

Definition at line 34 of file strategy.h.

The documentation for this class was generated from the following files:

- src/cpp/backend/header/strategy.h
- src/cpp/backend/strategy.cc

7.20 User Class Reference

User class for registration of new user and login if user already registered.

`#include <user.h>`

Public Member Functions

- User ()

  Contructor.
- void ReadSignUpForm ()

  Reading Sign up detail fields filled by user.
- void ReadLoginForm ()

  Reading login detail form filled by user.
- void SelectUserDetail ()

  Select user's email id and password from database.
- void LoginUser ()

  Login user if entered email and password matched with values in database.
- void SignUpUser ()

  Sign Up new user. If already registered then show error message.
- void **ConfirmUser** ()
- void **SetPassword** ()
- string Time ()

  For finding current time using boost library.
- ∼User ()

  Destructor.

Protected Attributes

- Database database
- ReadInputField **readField**
- SendMail **sendMail**
- STRING_VEC emailID
- STRING_VEC password
- STRING_VEC regKey
- STRING_VEC vecTemp
- string userEmailID
- string userPassword
- string retypePassword
- string currentTime
- string sessionID
- string **key**
- string msg
- string temp
- string where

7.20.1 Detailed Description

User class for registration of new user and login if user already registered.

Definition at line 27 of file user.h.

7.20.2 Member Function Documentation

7.20.2.1 void User::LoginUser ( )

Login user if entered email and password matched with values in database.

Matching user details with values in database

< If Email ID valid

< If Password Correct

< If Password Incorrect

< If Email ID invalid

Definition at line 71 of file user.cc.

7.20.3 Member Data Documentation

7.20.3.1 string User::currentTime [protected]

Currennt Time

Definition at line 43 of file user.h.

7.20.3.2 Database User::database [protected]

DataBase

Definition at line 30 of file user.h.

7.20.3.3 STRING_VEC User::emailID [protected]

Email ID as vector

Definition at line 34 of file user.h.

7.20.3.4 string User::msg [protected]

Confirmation Link Key Store error/info Message

Definition at line 43 of file user.h.

7.20.3.5 STRING_VEC User::password [protected]

password as vector variable

Definition at line 34 of file user.h.

7.20.3.6 STRING_VEC User::regKey [protected]

Registration key

Definition at line 34 of file user.h.

7.20.3.7 string User::retypePassword [protected]

For reading password

Definition at line 39 of file user.h.

7.20.3.8 string User::sessionID [protected]

Session Key

Definition at line 43 of file user.h.

7.20.3.9 string User::temp [protected]

Temporary variable

Definition at line 43 of file user.h.

7.20.3.10 string User::userEmailID [protected]

For reading emailID in text field

Definition at line 39 of file user.h.

7.20.3.11 string User::userPassword [protected]

For reading password in text field

Definition at line 39 of file user.h.

7.20.3.12 STRING_VEC User::vecTemp [protected]

Vector temp variable

Definition at line 34 of file user.h.

7.20.3.13 string User::where [protected]

where clause

Definition at line 43 of file user.h.

The documentation for this class was generated from the following files:

- src/cpp/frontend/header/user.h
- src/cpp/frontend/user.cc

Chapter 8

File Documentation

8.1 src/cpp/backend/arrangerollno.cc File Reference

Function definition of ArrangeRollNo class.

```
#include "header/arrangerollno.h"
```

8.1.1 Detailed Description

Function definition of ArrangeRollNo class.

Version

0.8

Date

Sunday 31 March 2013 06:50:00 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file arrangerollno.cc.

8.2 src/cpp/backend/datesheet.cc File Reference

Definition of member functions of Datesheet class.

```
#include "header/datesheet.h"
```

8.2.1 Detailed Description

Definition of member functions of Datesheet class.

Version

0.8

Date

Sunday 31 March 2013 09:39:11 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file datesheet.cc.

8.3 src/cpp/backend/expandrollno.cc File Reference

Function definition of ExpandRollNo class defintion.

```
#include "header/expandrollno.h"
```

8.3.1 Detailed Description

Function definition of ExpandRollNo class defintion.

Version

0.8

Date

Sunday 31 March 2013 06:49:23 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file expandrollno.cc.

8.4 src/cpp/backend/header/arrangerollno.h File Reference

ArrangeRollNo class for arranging roll no, sorting, and removing roll nos that are not required for seating plan.

```
#include "expandrollno.h"
```

Classes

- class ArrangeRollNo

  Arrange roll no class for sorting, removing redundant roll nos and also excludeing roll nos that are not for seating plan.

8.4.1 Detailed Description

ArrangeRollNo class for arranging roll no, sorting, and removing roll nos that are not required for seating plan.

Version

0.8

Date

Sunday 31 March 2013 06:33:39 IST
Compiler g++

Author

Mandeep Kaur, `meghasimak@gmail.com` License GNU General Public License

Copyright

Definition in file arrangerollno.h.

8.5 src/cpp/backend/header/datesheet.h File Reference

Datesheet class definition.

```
#include "arrangerollno.h"
```

Classes

- class DateSheet

  DateSheet class for reading datesheet detail, class detail and create datesheet.out file for seating plan.

8.5.1 Detailed Description

Datesheet class definition.

Version

0.8

Date

Sunday 31 March 2013 09:38:54 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Definition in file datesheet.h.

8.6 src/cpp/backend/header/expandrollno.h File Reference

ExpandRollNo class declaration for expanding roll nos.

```
#include "readinput.h"
```

Classes

- class ExpandRollNo

8.6.1 Detailed Description

ExpandRollNo class declaration for expanding roll nos.

Version

0.8

Date

Sunday 31 March 2013 05:45:55 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Definition in file expandrollno.h.

8.7 src/cpp/backend/header/readinput.h File Reference

Declaration of ReadInput class for reading input details from file.

```
#include "header.h"
```

Classes

- class ReadInput

  Class for reading input files ie reading roll no details, class detail, room detail, etc.

8.7.1 Detailed Description

Declaration of ReadInput class for reading input details from file.

Version

0.8

Date

Sunday 31 March 2013 02:33:49 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file readinput.h.

8.8 src/cpp/backend/header/report.h File Reference

Report class definition.

```
#include "seatplan.h"
```

Classes

- class SeatPlan

  SeatPlan CLass for generating seating plan.

8.8.1 Detailed Description

Report class definition.

Version

0.8

Date

Wednesday 03 April 2013 05:05:51 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file report.h.

8.9 src/cpp/backend/header/seatplan.h File Reference

Declaration of SeatPlan class for generating seating plan.

```
#include "readinput.h"
```

Classes

- class SeatPlan

  SeatPlan CLass for generating seating plan.

8.9.1 Detailed Description

Declaration of SeatPlan class for generating seating plan.

Version

0.8

Date

Tuesday 02 April 2013 02:05:39 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file seatplan.h.

8.10 src/cpp/backend/header/strategy.h File Reference

Strategy class for checking selected strategy is valid for seating plan or not.

```
#include "seatplan.h"
#include "../../header/database.h"
```

Classes

- class Strategy

  Strategy class to check selected strategy is valid or not for seating plan.

8.10.1 Detailed Description

Strategy class for checking selected strategy is valid for seating plan or not.

Version

0.8

Date

Tuesday 02 April 2013 02:09:24 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file strategy.h.

8.11 src/cpp/backend/main.cpp File Reference

Main Method.

```
#include "header/datesheet.h"
#include "header/strategy.h"
```

Functions

- int main (void)

  main function for declaring object of User class to register new user.

8.11.1 Detailed Description

Main Method.

Version

0.8

Date

Sunday 31 March 2013 08:13:40 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file main.cpp.

8.11.2 Function Documentation

8.11.2.1 main (void)

main function for declaring object of User class to register new user.

main function for declaring object of user class to register new user.

Definition at line 20 of file main.cpp.

8.12 src/cpp/backend/readinput.cc File Reference

Function Definition of ReadInput Class.

```
#include "header/readinput.h"
```

8.12.1 Detailed Description

Function Definition of ReadInput Class.

Version

0.8

Date

Sunday 31 March 2013 06:48:51 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers `https://github.com/GreatDevelopers`

Definition in file readinput.cc.

8.13 src/cpp/backend/report.cc File Reference

Report Class's menber function definition for generating report.

```
#include "header/report.h"
```

8.13.1 Detailed Description

Report Class's menber function definition for generating report.

Version

0.8

Date

Wednesday 03 April 2013 05:03:54 IST
Compiler g++

Author

Mandeep Kaur, `meghasimak@gmail.com` License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers `https://github.com/GreatDevelopers`

Definition in file report.cc.

8.14 src/cpp/backend/seatplan.cc File Reference

Function definition of SeatPlan Class.

```
#include "header/seatplan.h"
```

Functions

- void **error_handler** (HPDF_STATUS error_no, HPDF_STATUS detail_no, void ∗user_data)

Variables

- jmp_buf **env**

8.14.1 Detailed Description

Function definition of SeatPlan Class.

Version

0.8

Date

Tuesday 02 April 2013 02:44:21 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file seatplan.cc.

8.15 src/cpp/backend/strategy.cc File Reference

Function definition of Strategy class.

```
#include "header/strategy.h"
```

8.15.1 Detailed Description

Function definition of Strategy class.

Version

0.8

Date

Tuesday 02 April 2013 03:02:04 IST
Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com License GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file strategy.cc.

8.16 src/cpp/frontend/bakaplan.cc File Reference

Main Class for calling all pages and executing functions accoding to it flow

.

```
#include "header/bakaplan.h"
```

8.16.1 Detailed Description

Main Class for calling all pages and executing functions accoding to it flow

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Definition in file bakaplan.cc.

8.17 src/cpp/frontend/bakaplan_main.cpp File Reference

Baithne Ka Plan

.

```
#include "header/bakaplan.h"
```

Functions

- int **main** (void)

8.17.1 Detailed Description

Baithne Ka Plan

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Definition in file bakaplan_main.cpp.

8.18 src/cpp/frontend/database.cc File Reference

Definition of functions for database connectivity and accessing database.

```
#include "header/database.h"
```

8.18.1 Detailed Description

Definition of functions for database connectivity and accessing database. g++

Author

Mandeep Kaur, meghasimak@gmail.com GNU General Public License

Copyright

Definition in file database.cc.

8.19 src/cpp/frontend/header/bakaplan.h File Reference

Main class for calling all other classes

.

```
#include "header.h"
#include "project_detail.h"
```

Classes

- class BaKaPlan

  Main Project class.

8.19.1 Detailed Description

Main class for calling all other classes

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Definition in file bakaplan.h.

8.20 src/cpp/frontend/header/cgicc.h File Reference

Including header files required for using cgiic linrary.

```
#include <cgicc/CgiDefs.h>
#include <cgicc/Cgicc.h>
#include <cgicc/HTTPHTMLHeader.h>
#include <cgicc/HTMLClasses.h>
#include <cgicc/HTTPRedirectHeader.h>
```

8.20.1 Detailed Description

Including header files required for using cgiic linrary. g++

Author

Mandeep Kaur, meghasimak@gmail.com GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file cgicc.h.

8.21 src/cpp/frontend/header/database.h File Reference

Declaring Database class for creating specific functions for accessing database.

```
#include "header.h"
#include "database_detail.h"
#include <mysql.h>
```

Classes

- class Database

  Declaring functions for database comectivity and accessing database.

8.21.1 Detailed Description

Declaring Database class for creating specific functions for accessing database. g++

Author

Mandeep Kaur, meghasimak@gmail.com GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file database.h.

8.22 src/cpp/frontend/header/page_structure.h File Reference

functions for HTML tags which are commonly used and icluding common css and js files

```
#include <iostream>
#include <string>
```

Classes

- class PageStructure

  Declaring functions for common html tags and adding css and javascript files.

8.22.1 Detailed Description

functions for HTML tags which are commonly used and icluding common css and js files

g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file page_structure.h.

8.23 src/cpp/frontend/header/sendmail.h File Reference

SendMail class declaration for sending mails to user for registration or reset password link as confirmation mail.

```
#include <iostream>
#include "jwsmtp/jwsmtp.h"
#include "sendmail_detail.h"
```

Classes

- class SendMail

  Class for sending mail to user for registeration and for changing password.

8.23.1 Detailed Description

SendMail class declaration for sending mails to user for registration or reset password link as confirmation mail. g++

Author

Mandeep Kaur, meghasimak@gmail.com GNU General Public License

Copyright

Definition in file sendmail.h.

8.24 src/cpp/frontend/header/user.h File Reference

Sign up and login module. Authenticate user for logging in or registration of new user.

.

```
#include "header.h"
#include "database.h"
#include "read_inputfield.h"
#include "sendmail.h"
#include "md5.h"
```

Classes

- class User

  User class for registration of new user and login if user already registered.

8.24.1 Detailed Description

Sign up and login module. Authenticate user for logging in or registration of new user.

. Compiler g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Definition in file user.h.

8.25 src/cpp/frontend/input_detail.cc File Reference

Definition of functions

.

```
#include "header/input_detail.h"
#include "header/filename.h"
```

8.25.1 Detailed Description

Definition of functions

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file input_detail.cc.

8.26 src/cpp/frontend/page_structure.cc File Reference

Definition of Page Structure class functions

.

```
#include "header/page_structure.h"
```

8.26.1 Detailed Description

Definition of Page Structure class functions

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers https://github.com/GreatDevelopers

Definition in file page_structure.cc.

8.27 src/cpp/frontend/project_detail.cc File Reference

Function Definition of project detail class

.

```
#include "header/project_detail.h"
```

8.27.1 Detailed Description

Function Definition of project detail class

. g++

Author

Mandeep Kaur, meghasimak@gmail.com
GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers `https://github.com/GreatDevelopers`

Definition in file project_detail.cc.

8.28 src/cpp/frontend/sendmail.cc File Reference

Definitions of functions for sending emails to user and developer.

```
#include "header/sendmail.h"
```

8.28.1 Detailed Description

Definitions of functions for sending emails to user and developer. g++

Author

Mandeep Kaur, `meghasimak@gmail.com` GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers
`https://github.com/GreatDevelopers`

Definition in file sendmail.cc.

8.29 src/cpp/frontend/set_password_main.cpp File Reference

Set Password for new user

.

```
#include "header/user.h"
```

Functions

- int **main** (void)

8.29.1 Detailed Description

Set Password for new user

. g++

Author

Mandeep Kaur, `meghasimak@gmail.com`
GNU General Public License

Copyright

Copyright (c) 2013, GreatDevelopers `https://github.com/GreatDevelopers`

Definition in file set_password_main.cpp.

8.30 src/cpp/frontend/user.cc File Reference

Function definition of user class. See header file for more detail.

```
#include "header/user.h"
#include "header/fieldname.h"
```

8.30.1 Detailed Description

Function definition of user class. See header file for more detail. g++

Author

Mandeep Kaur, meghasimak@gmail.com GNU General Public License

Copyright

Definition in file user.cc.