

重定ゼミ

金曜 5限

重定先生

卒業制作論文
～ P e r l を使用した
アルバイト求人情報検索システム～

国際文化学部 国際文化学科

4 - G 0 1 G 0 6 1 7

秦 祐子

目次

- 1 . はじめに
- 2 . 目的
- 3 . システム概要
 - 3 - 1 . 学生用システム
 - 3 - 2 . 企業用システム
 - 3 - 3 . 管理者用システム
- 4 . システムの構成
 - 4 - 1 . CGIについて
 - 4 - 2 . Webデータベースについて
 - (1) リレーショナルデータベースとは
 - (2) SQLとは
 - 4 - 3 . 開発環境について
 - (1) Perlについて
 - (2) Webサーバ
 - (2) MySQLについて
 - (3) DBIとDBD
- 5 . システムの流れ
 - 5 - 1 . 学生用システムの流れ
 - (1) 全件検索
 - (2) 条件検索
 - (3) 検索結果表示
 - (4) 詳細表示
 - (5) 応募フォーム
 - 5 - 2 . 企業用システムの流れ
 - (1) 登録フォーム
 - (2) ログイン画面
 - (3) 企業情報変更・更新
 - (4) 求人情報追加
 - (5) 求人情報変更・更新
 - 5 - 3 . 管理者用システムの流れ
 - (1) ログイン画面

- (2) 企業情報の追加・更新・削除
- (3) 求人情報の追加・更新・削除

6 . プログラムの配置と解説

- (1) プログラムの配置情報
- (2) オブジェクト・メソッドの解説

7 . 考察・課題

8 . 謝辞

9 . 参考文献

1. はじめに

大学生にもなると、小遣い稼ぎや親の経済的負担を軽くするためなどその目的は様々であるが、アルバイトをしている学生の方が多いだろう。学生はよく遊ぶ。遊ぶにはやはりお金が必要だということで、アルバイトをするのである。実際に、様々な場面で実施されているアンケート結果等を見ても、かなりの割合で学生がアルバイトの経験があると答えているのが分かる。そこから多くの学生が日々、よりよい求人情報を探し求めているのではないか、ということが想像できる。

法政大学でも、企業から寄せられたアルバイト求人情報を学生に提供している。といっても、企業からの情報が就職課の方の手によって掲示板に貼り出され、それを見た学生が紙に記載されている連絡先に直接応募の連絡をするという実にアナログな仕組みである。その実態を先生から聞かされた時、インターネットの仕組みや様々なシステムについて学んでいた私は、違和感を覚えた。インターネットの普及したこの時代にあまりに似つかわしくない。

私はそれまでその掲示板を利用したこともなければ、存在すら知らなかった。実際、学生の間での認知度も低いと思われる。またそれがあると聞いてもなかなか足を運ぼうという気は進まない。掲示板の貼り出しとなると、やはり情報量も限られてしまうだろうし、わざわざ足を運ぶのも時間がかかるし、面倒だと感じるのだ。それに、今ではいろいろな雑誌やインターネットの検索サイトなど、情報源が豊富にあるので、わざわざ足を運ぶ必要がない。

近年インターネットの急速な普及により、Web上で合理的に求人情報を得ることも可能になった。書店や新聞の折り込み広告などに、求人広告及や求人情報誌があるが、種類が多い上に掲載されている情報数も多いので、チェックする時間をかなり要する。また掲載スペースの都合で内容説明が不十分である場合もある。しかし、インターネットの利用目的の多様化により、実に様々な情報を「いつでも」「どこでも」「必要なものだけ」「瞬時」に検索し、得られるようになった。この情報検索の機能は、はじめから条件を提示して検索することができる、インターネットの最大の特徴であろう。

この流れで、仕事情報検索や仕事紹介といった種のサイトが増えている。自分の希望する条件から仕事を検索したり、見つけた会社に直接応募や登録したりできるものだ。情報量が多く業種や職種、勤務地や給与までかなり細かく検索できるし、直接企業の方にアクセスできるという点でとても便利で利用しやすい。また体験談や画像・映像などもあり、情報を色んな面から、しかも瞬時に得られるのも利点だ。

このようにインターネットが普及し、大変機能的なサイトが増えている中で、本校での情報提供のやり方は様々な手間がかかる上に、学生としても利用しにくいということで、やはりあまり有効ではない。就職課の方と企業側とのやりとりや貼り出しの労力、また学生が足を運ぶ手間などの他にも、情報発信から掲示されるまでの時間のギャップといった改善すべき問題点等もいくつかあげられる。

そこで、これらの問題を解決するのが、この「求人情報検索システム」である。今行われている手作業部分をC G Iプログラムによってシステム化し、企業からの求人情報の掲載、学生の情報検索・仕事の応募、また情報の管理までをW e b上で一括して行えるようにすることで、学生、企業、学校の事務員、全ての者の手間や時間を軽減する。インターネットを介して一つのデータベースを共有することで、これを実現できると考えたのだ。

本論文は、以上の動機から作成した「アルバイト求人情報検索システム」について、その目的やシステム構成、概要等について動作例の図や説明を交えながら展開していく。

2 . 目的

前項で述べたとおり大学のアルバイト求人情報提供のシステムでは、手間と時間がかかり、それによる問題点が多くある。はじめにその手間と問題点を整理しておく。

< 手間・労力 >

学生が掲示板まで足を運ぶ時間と手間

応募する連絡先をメモし、各自別の場所から連絡をとる手間。

就職課の方の掲示物準備と貼り出しの手間。

ちょっとした変更でもその都度学校へコンタクトする手間。

< 問題点 >

新しい情報を得るために、学生はこまめに足を運ばなくてはならない。

管理やスペースの関係上、どうしても情報数に限りが出てくる。

他に便利な情報源があるので、学生の利用が少ない

学校側での情報管理がしにくい。紙なのでなくしたり、似たような情報があると見分けにくかったりと情報が多くなればなるほど混乱する。

新情報の発信から、情報掲載までに時間がかかる。急を要するものには対応できない。

掲示板全体から希望にあった情報を探すのには時間がかかり、全てを見るのも大変。

以上、改善すべき問題点である。このいかにも手間も時間もかかるこの状況を改善しなくてはならない。

そこで以下に大きく三つ、本システムの目的を示す。

- (1) 学生、企業、学校側 (就職課) 三者の労力 (手間) の削減
- (2) 新しい情報が発信されてから学生に届くまでの時間的ギャップの解消
- (3) 情報管理の簡易化

また、目的の先に見えるそれぞれの主なメリットをまとめる。

2 - 1 . 学生のメリット

- (1) 「 掲示板まで足を運ぶ 」 という時間の浪費とその労力が削減され、時間と場所を選ばずいつでも情報を検索できる。
- (2) W e b で見て応募を希望するなら、その場で応募フォームに入力するだけで簡単にその企業にアクセス (応募) できる。
- (3) 各企業が直接データを更新することで、常に新しい情報を閲覧できる。

2 - 2 . 企業側のメリット

- (1) W e b上から自社の企業データ変更、及び求人情報の追加・更新が可能なので、大学を通す手間を削減。常に新しい情報をいち早く提供していくことができる。
- (2) 大学側を通す時間と手間が削減されることで、新情報の依頼から掲示板によって学生へ情報が伝えられるまでの時間的ギャップを解消できる。
- (3) 掲載情報の内容確認がしやすい。(いちいち学校に聞かなくてもよい)

2 - 3 . 大学側のメリット

- (1) 情報掲載を依頼する企業や、その求人情報のデータ件数が増えても管理が容易である。
- (2) 企業から依頼を受けてから、掲示板に貼り出すまでの手間が省ける。

3 . システム概要

このアルバイト求人情報検索システムは、あらかじめ登録された企業、また仕事を探す学生が、webブラウザからデータベースを直接操作することで、常に情報更新や検索、仕事応募などが一括してできるシステムである。

Web データベースを利用しデータを Web サーバに置くことで、ユーザが Web ページにアクセスする感覚で、データ操作することを可能にした。

本システムは、「学生用システム」「企業用システム」「管理者用システム」のおおきく3つに分けられる。このシステムこれまで企業と学生との間で情報の管理、伝達をしていた学校の就職課の方が管理者となる。よって、以降説明する管理者とは、就職課の事務の方にあたる。ここでは、それぞれのシステムにおける機能について説明する。

< 3 - 1 . 学生用システム >

学生用システムは、検索ページから現時点での最新のアルバイト求人情報を閲覧検索、そして応募まで一括してできるようになっている。場所や、時間を選ばず常に新しい情報を得られ、応募フォームから応募できることが大きな利点である。

以下に主な機能を示す。

求人情報全件表示

全件表示のボタンを押せば、学校側に寄せられている求人情報が全て閲覧できる。表示は、仕事情報を登録する際に振り分けられるID番号の昇順になっている。「詳細表示」ボタンから更に詳しい仕事情報や、企業情報を確認できる。

条件検索

「職種」「勤務地」「給与」などの条件を選択もしくは入力することで、それぞれの条件に合った仕事のみを瞬時に検索し、表示することができる。ここでは、給与の高い順に検索結果表示がなされるようになっている。条件検索で選択されたものの中では、より給与の高い仕事に興味を持つだろうと、学生側の心理を考えてのことである。この結果表示画面にも「詳細表示」機能は設けてある。

詳細表示

全件表示、また検索結果の中からそれぞれの仕事情報に関する、さらに詳しい仕事内容や、そのアルバイト先の企業情報を表示する。仕事情報を格納するテーブル(jobinfo_t)の中の企業ID番号と、企業情報のテーブル

(jobprof_t)のID番号を参照することで、その特定のフィールドを呼び出している。なお、この詳細表示画面からWeb応募フォームへリンクできるようになっている。

Web応募フォーム

このWeb応募フォームに自己プロフィールや志望動機等を入力し、送信ボタンをおすだけで相手企業へメールにて応募フォームが送られる。詳細を見て応募を決意した学生は、時間を気にすることなく送信(応募)することができる。

< 3 - 2 . 企業用システム >

本システムのメイン機能の一つである、Webからのデータ操作(追加、更新)を実現しているのがこの企業用システムである。ID,パスワードによるログイン機能を設けることで、学校を通さずに各企業が直接データベースを操作し、必要に応じて情報を更新していくことを可能にした。これによって、学生も常に新しい情報をいち早く得られるのである。

会員登録フォーム(企業情報登録)

情報を掲載するには、どの企業も会員登録が必要である。この会員登録フォームに企業情報等を入力し、登録ボタンをおすと、データベースにその内容が追加される。後に管理者から登録完了に関するお知らせと、ID、パスワードが送られてくる。管理者からの連絡がなければまだ情報掲載はできない。

ログイン機能

求人情報の追加や更新をするには、企業ごとにログインが必要である。会員登録フォームで登録したパスワードと、登録後に送られてくるIDでログインする。

また、登録した企業情報の更新をするにも同じくログインが必要である。ただし、ログインしても情報の削除はできないようになっている。データの削除ができるのは管理者のみなので、削除に関しては管理者に直接依頼する必要がある。

企業情報更新

会員登録の際に入力した企業情報を変更することができる。ただし、更新にはログインが必要。ログインが成功すると、情報変更フォームが表示され、更新ボタンを押すと、データベースが更新される。

求人情報追加

ここで、求人情報を追加する。基本的に各企業はいくつでも求人情報を追加することができる。データ追加にもログインは必要である。ログインが成功し、全ての項目を正常に入力できれば、データは追加される。

求人情報更新

以前に登録された求人情報を変更・更新することができる。ログインに成功すると、その企業のもつ求人情報が全て表示されるので、その中から変更したいものだけを選んで、新しい情報を入力し更新する。更新は一件ごとしかできない。

< 3 - 3 . 管理者用システム >

これまで企業と学生の中の仲介役となっていた就職課の方が本システムの管理者となる。管理者は、求人情報及びシステム全体を監視し、不適切なものがあれば削除するなどして対応をとる。

管理者用システムでは、全てのデータを操作できるようになっている。企業用システムではデータの削除はできないようになっており、削除できるのは管理者のみである。管理管理者のページへは、あらかじめ管理者の決めたIDとパスワードのみログインし、アクセスすることができる。

管理者ログイン

管理者ページへのログイン機能。あらかじめ決められた管理者用のID・パスワードでログインを実行する。ログインが成功すれば、シークレットページである管理者用ページにアクセスすることができる。

求人情報全件表示・更新・削除

求人情報が格納されているテーブル(jobinfo_t)のデータ操作機能である。ログインした画面から、企業が追加・更新している求人情報全てに関して、閲覧、変更・更新または削除ができる。削除に関しては、その求人情報のID番号を指定するだけの簡単な操作で削除が実行される。

企業情報追加・更新・削除

登録された企業情報を格納するテーブル(jobprof_t)のデータ操作機能である。上記の機能と同様、ログインした画面から企業情報データの閲覧、変更・更新、または削除ができる。企業ID番号を指定して削除を実行する。

4 . システムの構成

本システムは、「CGI」というユーザがブラウザ（web上）でプログラムを実行するための仕組みと、Webからのデータベース操作を可能にする「Webデータベース」を利用することで実現している。

本項では、この「CGI」と「Webデータベース」に関する説明と、開発環境について解説する。

4 - 1 . CGIとは

Common Gateway Interface の略。簡単に言えば、「www ブラウザと www サーバマシン上のプログラムとの間でデータのやりとりを行う仕組みのこと」である。普通のホームページは、ブラウザの要求に対して送られるファイルを表示していただいただけであるが、CGIを仕様したものは、ブラウザの要求に対して www サーバが別のプログラムを動かし、何かしら処理が行われた実行結果が表示される。このサーバ上で実行されるプログラムをCGIプログラムと呼ぶ。正否の判定を伴うクイズやゲーム・アクセスカウンタ・アンケート・掲示板・データベース検索などといったホームページがその例である。その中で検索条件を入力し、検索を実行、結果を表示することができる、データベース検索機能が私の制作するシステムの基本となる。

また、CGIの特徴を以下に挙げる。

< 利点 >

動的に変化する

プログラムによってページに変化をもたせることができる。

見る側の環境に左右されない

ホームページを見る側が使っているパソコン、OS、ブラウザの種類にかかわらず同じ内容を見られる。

プラグインが不要

見る側のパソコンに特別なプラグインをダウンロードしてインストールする必要はない。

< 欠点 >

プログラミングの知識が必要

誰でもすぐできるというのではなく、完成するまでに何度も何度も試行錯誤する時間が必要である。

完成までの手順が複雑

HTMLのようにその場で確認できるわけでもなく、ホームページ作成ソフトのような便利なものもない。

サーバマシンの負荷が上がる

サーバマシンの CPU 使用率やメモリ消費量、ハードディスクアクセス頻度などが高まる。

4 - 2 . W e b データベースについて

求人情報、企業情報などのデータを格納するためのデータベースが必要になる。しかもブラウザから、データベースの検索（取り出す）、変更、削除などの操作が直接できなければならない。それを可能にするのが **Web データベース** である。データを Web サーバに置くことで、ユーザは Web ページにアクセスする感覚で、データを取り出したり入力したりできる。このとき Web サーバ側には、**DB アプリケーション** と **ミドルウェア**（データベースと Web サーバをつなぐインターフェース）、**Web サーバアプリケーション** が必要になる。今回は、Web サーバとして Apache を使い、Perl を通じてデータベースの操作を行う。（Apache と Perl に関しては以降で解説する）

（ 1 ）リレーショナルデータベースとは

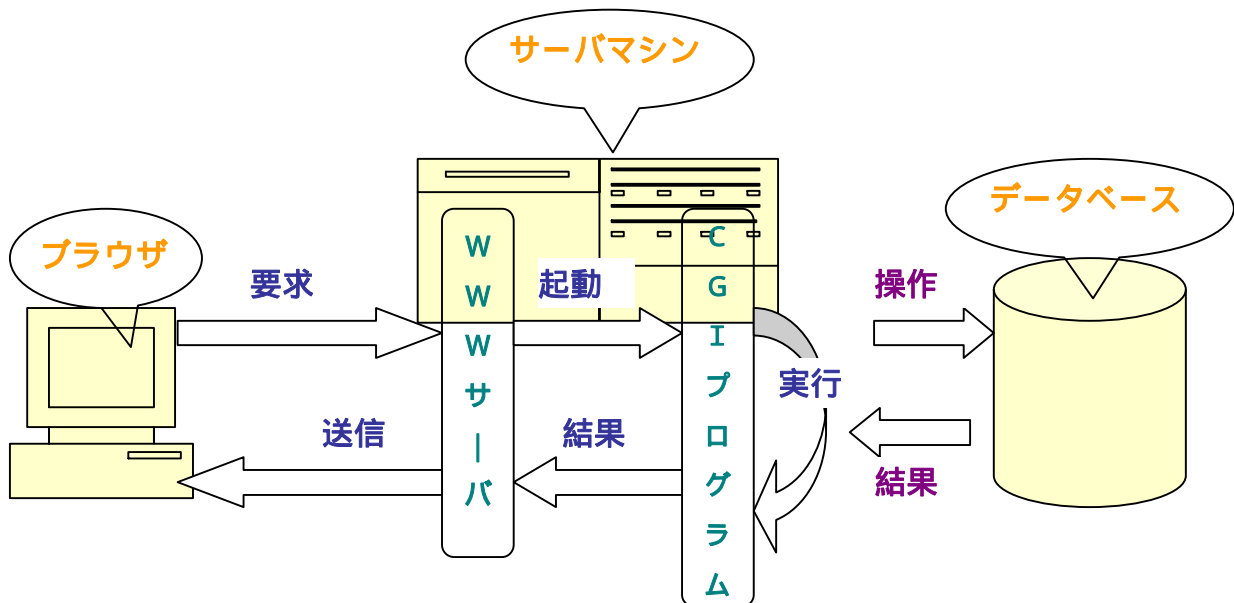
そもそもデータベースには様々な種類があるが、現在もっとも普及しているのは **リレーショナルデータベース** である。この特徴は、表計算ソフト（例えばエクセル）のような **表（テーブル）** を持っており、この表の **行（レコード）** を一件分のデータとし、行を構成する **列（フィールド）** に実際のデータを格納する。また、フィールドの中でも、各レコードを 1 行ずつ識別することのできる特殊なフィールドを **主キー（プライマリキー）** と呼び、あるテーブルの主キーを他のテーブルのフィールドに持たせたものを **外部キー** という。必要であればテーブルを複数持たせ、テーブル間の関係を主キーと外部キーを使って集合論的に結びつけ、膨大な量のデータを簡単に処理させることができるのである。このようなデータベースの構造は **スキーマ** と呼ばれる。

また、リレーショナルデータベースを管理するシステムを **リレーショナルデータベース管理システム (RDBMS)** と呼ぶ。

（ 2 ）SQL とは

リレーショナルデータベースの基本的な使用方法を ISO（International Organization for Standardization）や ANSI（American National Standard Institute）で標準化されており、そのデータベースを操作するための言語を **SQL（Structured Query Language）** という。

ここに、CGI プログラムと Web データベースの構成図を示す。



4 - 3 . 環境開発について

(1) Perlについて

CGI プログラムに使われるスクリプト言語としては、C 言語、ASP、JSP、PHP などがあげられるが、その中でも今回のシステム制作では Perl を使用した。

Perl は、本来 CGI のために開発された言語であるわけではないが、現在の CGI プログラムに最も多く用いられている。しかも、CGI プロバイダのサーバマシンには必ずといってよいほどインストールされており、サーバから簡単に呼び出すことが可能である。また Perl はテキストファイル进行处理するのに適した言語であるといわれている。Perl でサポートされる配列やハッシュ、繰り返し処理や入出力機能、また正規表現といったものは、テキストファイル进行处理するのに非常に都合がよくできているのである。CGI で Perl が多く用いられているのは、CGI の内部の処理がテキストファイル进行处理することで事足りることが多いからだ。

今回制作する仕事検索システムでの処理は文字列検索の機能が主となるので、これに強い Perl を使用することを決めた。以下に Perl の利点を挙げる。

無料である

インターネットや書籍の付録CDで入手可能。

多くのコンピュータに移植されている

Unix, MS-DOS, FreeBSD, Linux, Windows, Mac OS など、どのOSのコンピュータで作った Perl スクリプトでも、そのまま別のコンピュータでも使える。

文字列処理に強い

一般的に Perl の実行速度は C 言語で作られたプログラムに比べると遅い。しかし、文字列操作においては C 言語より豊富で高度な機能をもつ。よって文字列の検索などはきわめて高速に行えるのである。

インタプリタ型である

Perl スクリプト (テキストファイル) を Perl というインタプリタに読み込ませてその場で実行させるので、コンパイルは必要ない。更にデバッガも付属しているので開発やデバッグが容易である。

これらのような特徴から Perl が CGI プログラムに向いているといえる。

(2) Web サーバ

CGI プログラムをサーバマシンに送る前に、パソコン用サーバソフトと Perl ソフトを使ってインターネットにつながずにパソコン上でテストをする。これには Windows 用サーバソフト、Apache for Win32 を使用する。Apache は元々 UNIX 系 OS の www サーバソフトで、これを採用しているプロバイダも数多く、高い実績を誇っている。インターネットにつながずにパソコンだけで CGI を動かすことができる利点は大きいと後々実感した。

(3) MySQL について

既に説明したとおり、データベースでデータを操作する際には、「SQL」という言語を使う。データの追加、削除、変更、検索などを行うだけでなく、データベースの構造やデータ型を定義する言語でもある。MySQL は RDBMS (リレーショナルデータベース管理システム) の 1 つで、他には PostgreSQL、Oracle などがある。MySQL の特徴を以下にあげる。

とにかく処理速度が速い

非常に多くのデータを扱えるので、広い範囲での応用が可能

C、Java、Perl、PHP、Ruby など様々なプログラミング言語と接続可能

多言語対応で、日本語は、SJIS、EUC とともにサポートしている

セキュリティ管理がしやすく、Web データベースとして最適である

自由にダウンロードして使用し、目的に応じて修正することができる

< SQLコマンド >

MySQL でデータを入力したり出力したりするには SQL コマンドを発行する。SQL は基本操作である「追加」「変更」「削除」「検索」に使う命令で構成されている。以下がその主な注意点である。

SQL コマンドに大文字・小文字の区別はない

最後は必ず「 ; 」を入れて Enter を押す

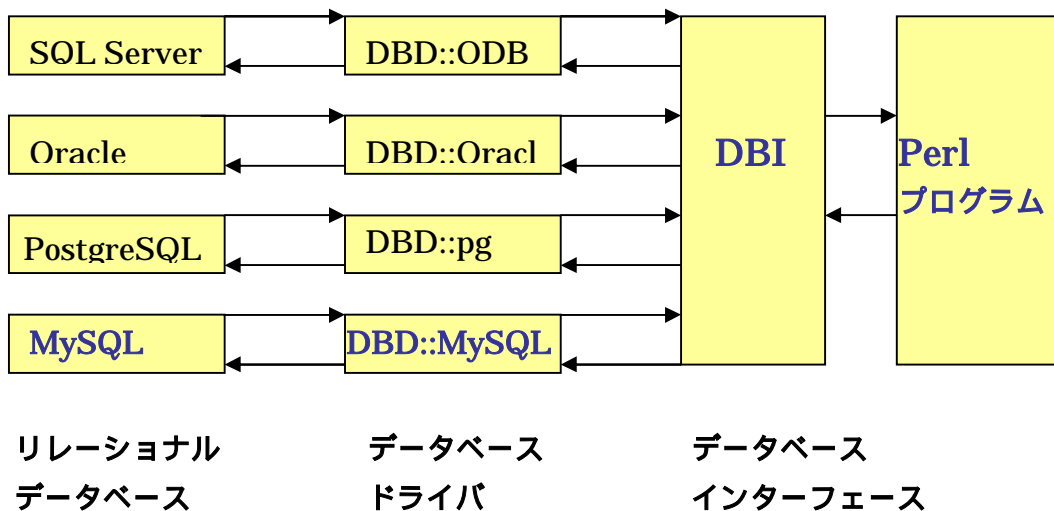
複雑なつづりに気をつける

アルファベット、数字、その他の記号は必ず半角で入力する

(3) DBI と DBD モジュール

Web ページからのデータベース操作をデータベースがサーバ上で直接行う事はできない。そこで、プログラミング言語とデータベースの橋渡しをするモジュールやドライバが、データベースからの操作結果を HTML の一部として Web サーバに渡す役割を果たしている。DBI はリレーショナルデータベースの種類にかかわらず統一的に扱うインターフェース・モジュールであり、これにより様々なリレーショナルデータベースの差異をほとんど気にせず、データベースを操作することが可能になる。また DBD は、プログラミング言語 (Perl) とデータベース (MySQL) を接続するためのドライバである。

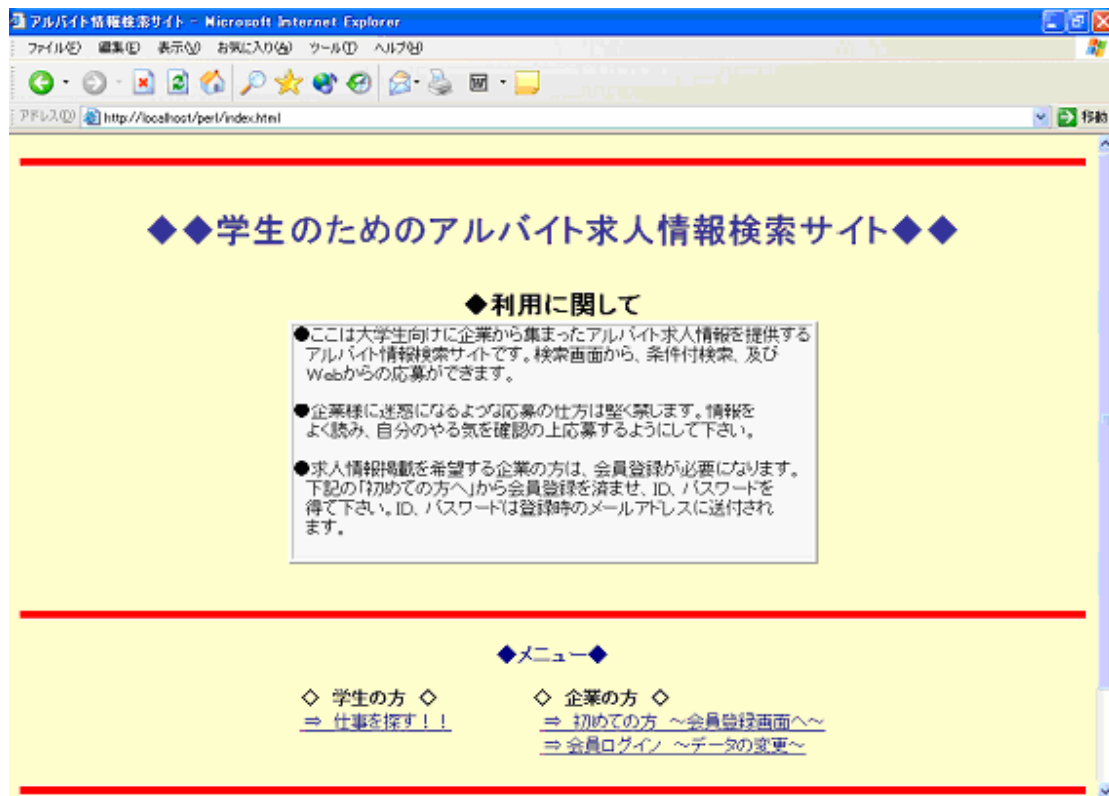
DBI と DBD の構成を下の図に示す。



5 . システムの流れ

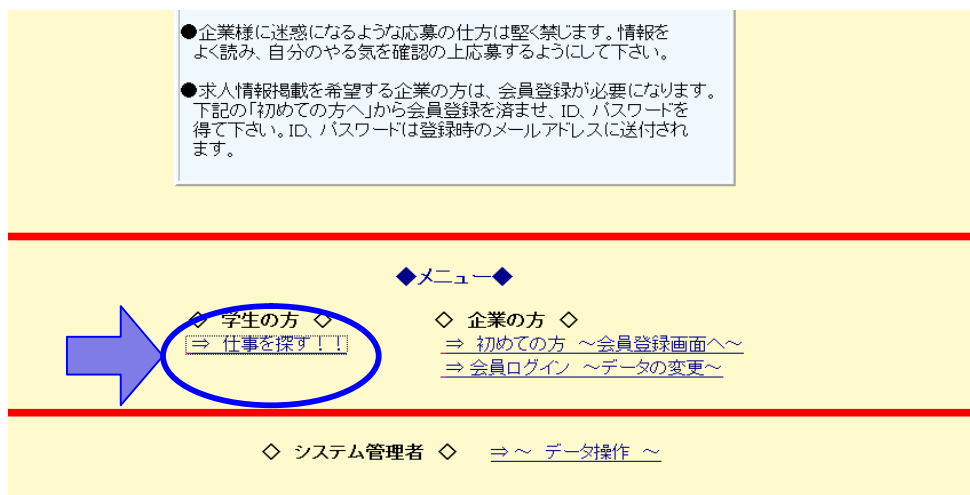
ここでは実際の各システムの流れを、動作例を入れながら説明する。

以下に示す図がトップページである。



5 - 1 . 学生用システムの流れ

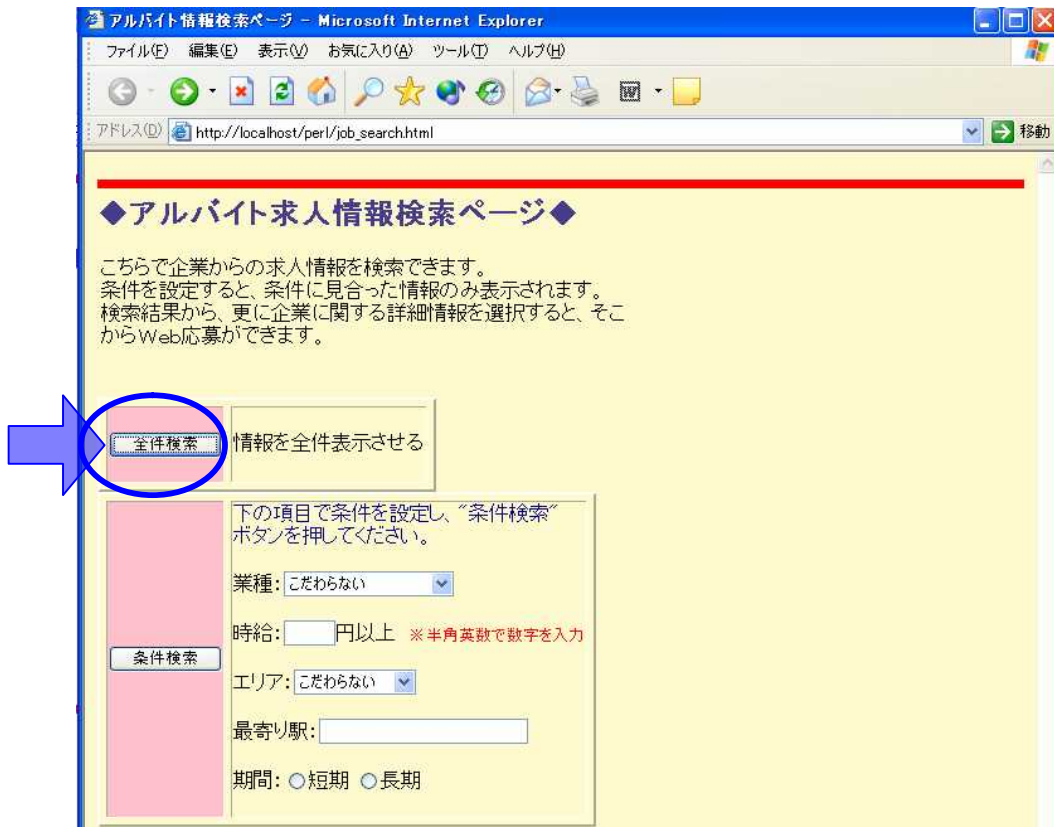
学生用のシステムの流れを説明する。学生用のシステムは、検索機能がメインとなっている。学生は好きな時間に、好きな場所から自由にアクセスし、情報を閲覧できる。トップページから、【仕事を探す!!】をクリックして、検索ページへとリンクする。



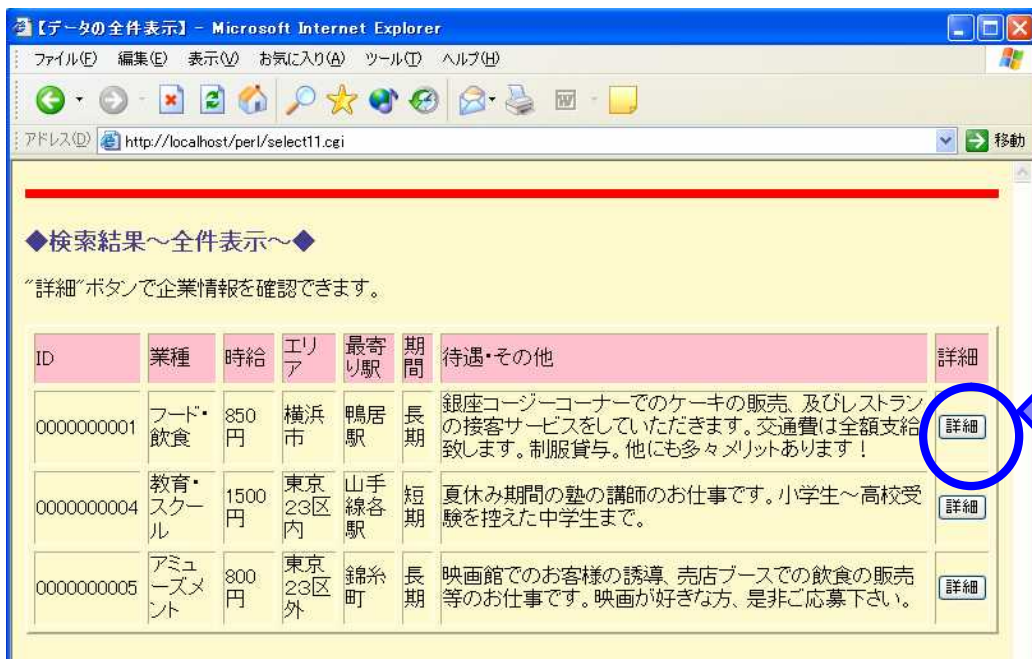
(1) 全件検索

登録されている求人情報を全件表示する。

全件表示ボタンをおす



ID番号の昇順でデータが全件表示される



各件ごとに詳細ボタンがあり、そこからもっと詳しい情報を見られる。

この詳細表示に関しては(3)で説明する。

(2) 条件検索

テーブルの中から自分の希望に合った情報を検索し、表示させる。

各項目の条件を設定、入力する。

◆アルバイト求人情報検索ページ◆

こちらで企業からの求人情報を検索できます。
条件を設定すると、条件に見合った情報のみ表示されます。
検索結果から、更に企業に関する詳細情報を選択すると、そこからWeb応募ができます。

全件検索 情報を全件表示させる

下の項目で条件を設定し、“条件検索”ボタンを押してください。

業種: フード・飲食

時給: 850 円以上 ※半角英数で数字を入力

エリア: 横浜市

最寄り駅:

期間: 短期 長期

条件検索

条件検索ボタンをおし、検索を実行する。

該当データが見つければ、 で設定した条件に合ったデータのみ表示される。

Perl【データの検索】 - Microsoft Internet Explorer

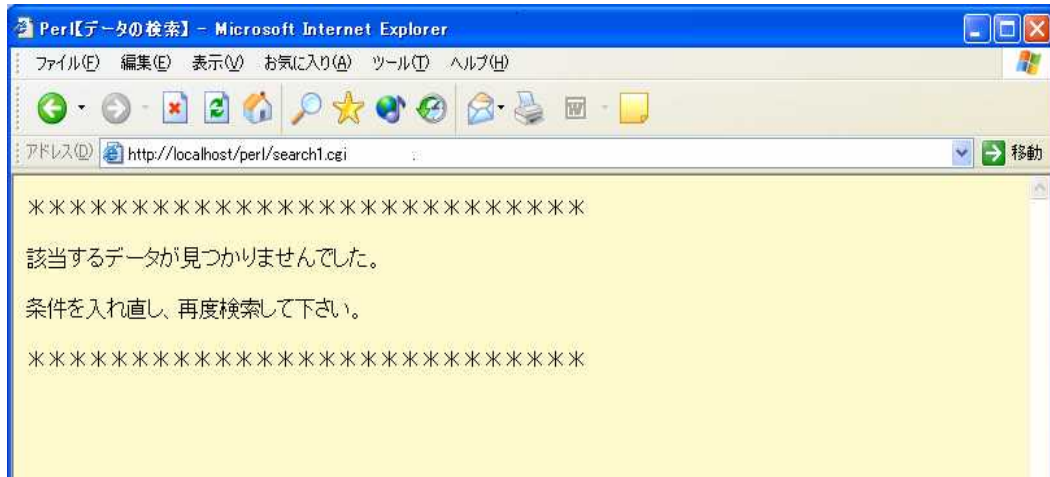
アドレス http://localhost/perl/search1.cgi

データの検索
検索結果を表示します

ID	職種	時給	エリア	最寄り駅	期間	待遇・その他	詳細
0000000001	フード・飲食	850 円	横浜市	鴨居駅	長期	銀座コージーコーナーでのケーキの販売、及びレストランの接客サービスをしていただきます。交通費は全額支給致します。制服貸与。他にも多々メリットあります！	詳細

各件ごとに詳細ボタンがあり、そこからもっと詳しい情報を見られる。
この詳細表示に関しては（３）で説明する。

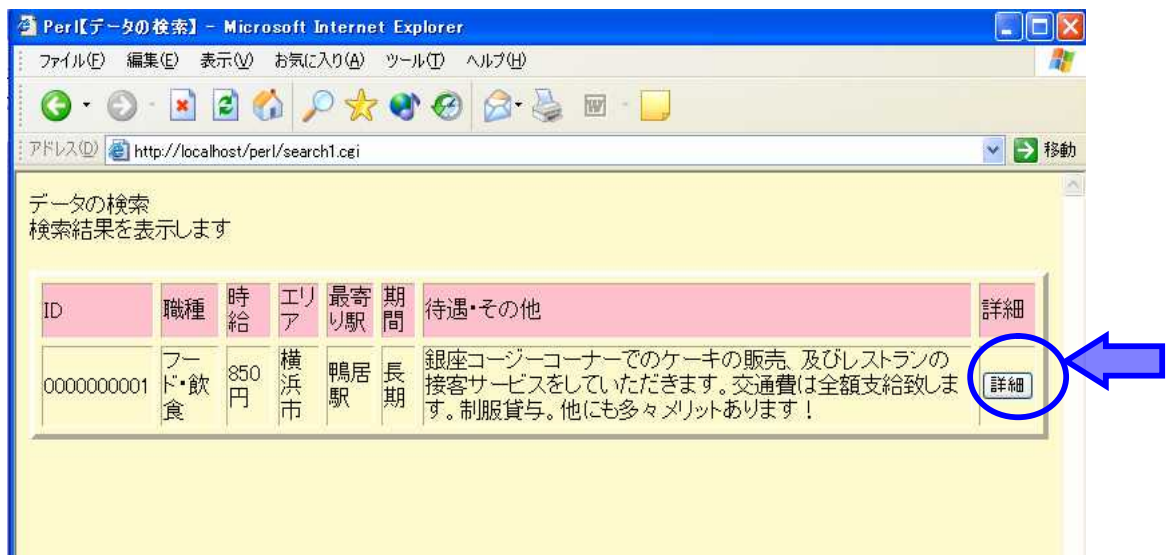
条件に合ったデータが一件もない場合は、「該当するデータが見つかりませんでした」とうエラーメッセージが表示される。



（３）詳細表示

検索結果の中から、特定の企業のプロフィールやその他詳しい情報を表示する。

検索結果の表の一番右端にある詳細ボタンをおす。



選択された求人情報を掲載している企業のプロフィール等が表示される。

◆詳細表示◆

選択されたアルバイト情報の詳細を表示します。
以下の内容を確認した上で、応募方法の指示に従い、応募
して下さい。尚、Web応募フォームからも応募できます。

ID	00000000001
会社名(店名)	銀座コージーコーナー鴨居店
所在地	神奈川県横浜市緑区鴨居
電話番号	0459333333
メールアドレス	cozy_kamoi@hotmail.com
時間帯・条件など	土日の午前、平日夕方～22時30分まで働ける方。
応募方法・備考	web応募フォームから応募してください。電話でも受け付けております。お気軽にお電話下さい。

Web応募フォーム

情報を見て応募を希望する場合は、下のWeb応募フォームというボタンをおす。web応募フォームについては次項で説明する。

(4) 応募フォーム

応募を決めた学生は、Web応募フォームから応募することができる。

詳細情報の下にあるWeb応募フォームボタンをおす。

◆詳細表示◆

選択されたアルバイト情報の詳細を表示します。
以下の内容を確認した上で、応募方法の指示に従い、応募
して下さい。尚、Web応募フォームからも応募できます。

ID	00000000001
会社名(店名)	銀座コージーコーナー鴨居店
所在地	神奈川県横浜市緑区鴨居
電話番号	0459333333
メールアドレス	cozy_kamoi@hotmail.com
時間帯・条件など	土日の午前、平日夕方～22時30分まで働ける方。
応募方法・備考	web応募フォームから応募してください。電話でも受け付けております。お気軽にお電話下さい。

Web応募フォーム

ボタンにリンクがはられており、Web応募フォームページへと移動する。

Web応募フォームに自分のプロフィールや志望動機等を入力する。

◆Web応募フォーム◆

以下の項目にご入力の上、送信ボタンを押してください。
こちらの情報は各会社のメールアドレスに送られます。

◆お名前 (必須):

◆フリガナ (必須):

◆性別 (必須):
 男性 女性

◆年齢 (必須):
 歳

◆ご住所 (必須):

◆電話番号 (必須) ※携帯or自宅:

◆メールアドレス (必須):

◆勤務希望時間(複数選択可):
 早朝 午前～昼 昼～夕方
 夕方～夜 深夜

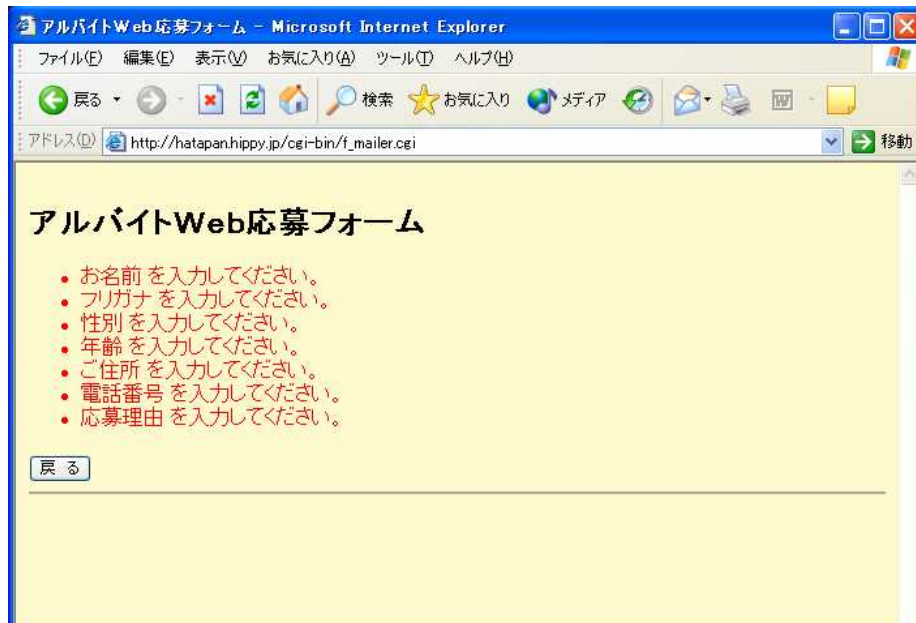
◆応募理由: (必須) ※100～150時程度

◆要望・質問等:

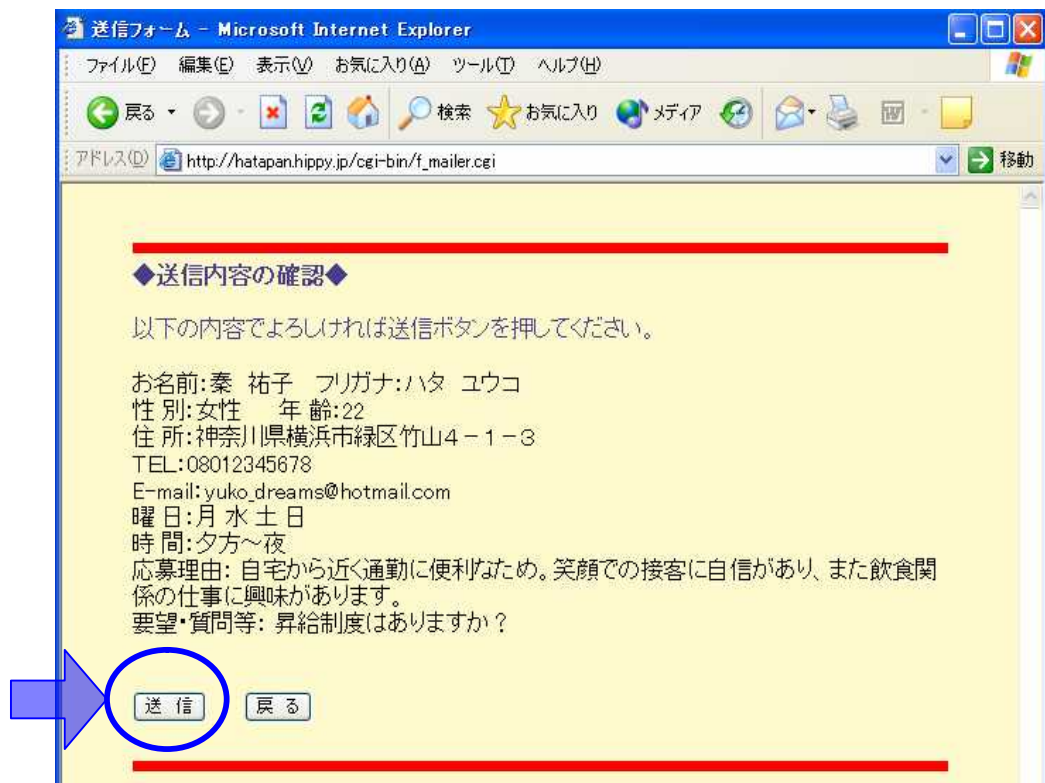
◆ 確認画面を出さずに送信

送信ボタンをおす。間違えればクリアボタンでクリアされる。

必須項目の入力もれがあると、エラーメッセージが表示される。



「確認画面を出さずに送信」のチェックボックスにチェックを入れずに送信ボタンをおすと、内容確認画面が表示される。



内容があていれば、そのまま送信ボタンをおす。間違えがあればバックボタンで戻って入力し直す。

正常に送信されれば、メッセージとともに送信完了画面が表示される。



応募フォームは各企業情報に登録されているメールアドレス宛てに送られる。
以下が試用で送られてきた応募フォームメールの本文部分である。

宛先: hatayuko1@hotmail.com
件名: アルバイト応募フォーム

フォームの送信

◆お名前 : 秦 祐子
◆フリガナ : ハタ ユウコ
◆性別 : 女性
◆年齢 : 22
◆ご住所 : 神奈川県横浜市緑区竹山4-1-3
◆電話番号 : 08012345678
◆EMAIL : yuko_dreams@hotmail.com
◆勤務可能日 : 月 水 土 日
◆勤務希望時間 : 夕方～夜
◆応募理由 : 自宅から近く通勤に便利のため。笑顔での接客に自信があり、また飲食関係の仕事に興味があります。
◆要望・質問 : 昇給制度はありますか？

5 - 2 . 企業用システムの流れ

企業用のシステムの流れを説明する。トップページに、企業の方向へのリンクを貼ってあるのでそこから利用が可能である。企業用のもとしては、初めて訪れた方用の【会員登録フォーム】と、既会員の方用の【データ更新ページ】とに分かれる。

(1) 登録フォーム

このサイトに求人情報を掲載するには会員登録が必要である。その登録をするのがこの登録フォームである。

各項目の指示に従って、正確に企業情報を入力する。

◆会員登録フォーム◆

こちらは、「学生のためのアルバイト情報検索サイト」に求人情報を提供して下さる企業様向け、会員登録フォームです。

御社の企業情報を正確にご入力下さい。情報に間違いがないかお確かめのの上 [送信ボタン]を押して下さい。

なお、こちらの情報は、アルバイト検索結果から選択された際に、会社プロフィール・仕事内容に関する詳細として表示されます。

◇会社名(店名): 法政オペレーションシステム

◇所在地: 東京都千代田区飯田橋 * * * *

◇電話番号: 0312345678
※-ハイフン無し。半角英数字で入力

◇メールアドレス: hosei@hatapan.jp

◇時間・条件など: ※全角100字まで
夕方から11時ころまで働ける学生の方。
土日祝日は朝から働ける方。夜勤もあります。

◇応募方法・備考: ※全角125字まで
web応募フォームからご応募下さい。電話での応募は受け付けておりません。質問などございましたら、メールで受け付けております。お気軽にご連絡下さい。

◇パスワード: 01hosei ※半角英数10字以内

登録 リセット

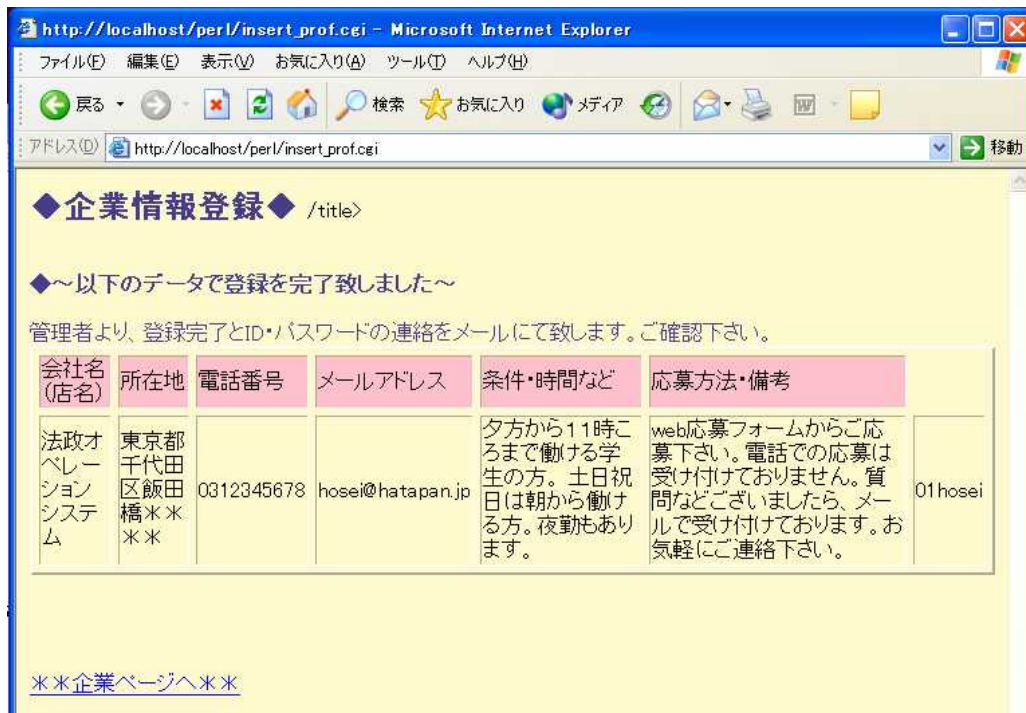
登録ボタンを押す。

正しく入力されていれば、正常にデータベースにデータが追加される。

入力漏れがあると、エラーメッセージが表示されるので、ブラウザのバックボタンで戻って入力しなおす。



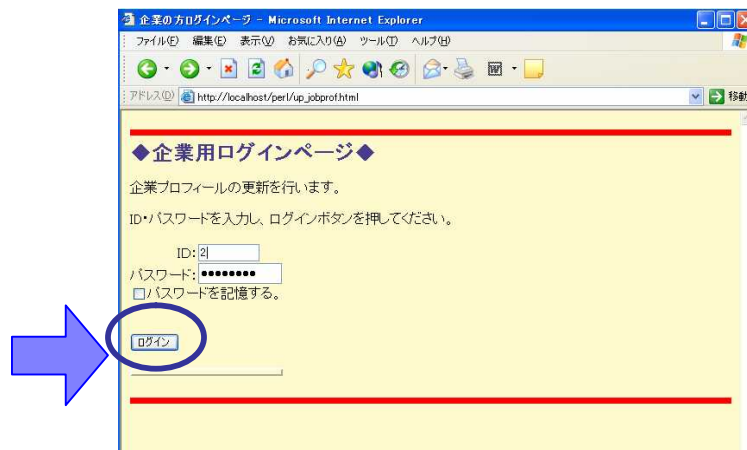
登録内容とともに登録完了のメッセージが表示される。



(2) ログイン画面

企業情報、また求人情報のデータ操作をする際には、ログインが必要である。

まず、登録時に管理者から送られたID・パスワードを入力する。



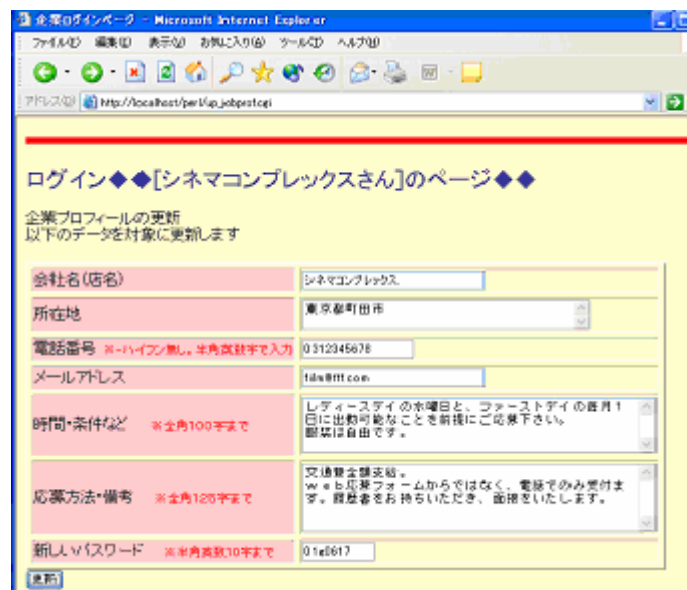
◆企業用ログインページ◆

企業プロフィールの更新を行います。
ID・パスワードを入力し、ログインボタンを押してください。

ID: 2
パスワード: *****
 パスワードを記憶する。

ログイン

正しく入力されれば、プログラムが起動し、情報更新フォームが表示される。



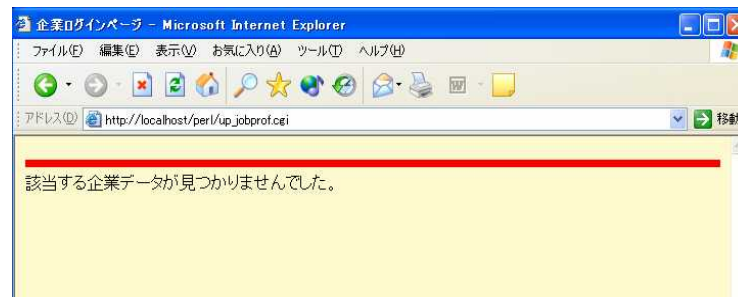
ログイン◆◆[シネマコンプレックスさん]のページ◆◆

企業プロフィールの更新
以下のデータを対象に更新します

会社名(店名)	シネマコンプレックス
所在地	東京都町田市
電話番号 ※ハイフン無し、半角英数字で入力	0312345678
メールアドレス	tds@tt.com
時間・条件 ※全角100文字まで	レディースデイの水曜日と、フォーストデイの毎月1日に出勤可能なことをご前提にご応募下さい。服装は自由です。
応募方法・備考 ※全角120文字まで	交通費全額支給。 Web応募フォームからではなく、電話でのみ受付です。履歴書をお持ちいただき、面接をいたします。
新しいパスワード ※半角英数10文字まで	01e0617

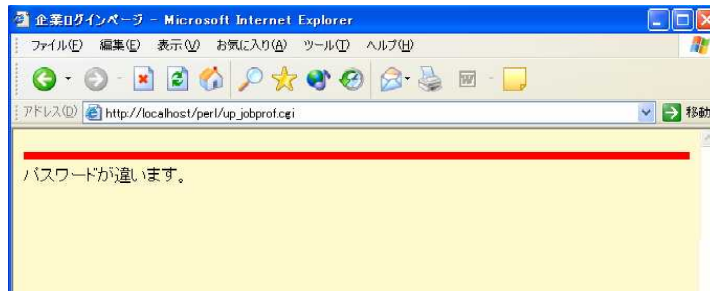
更新

なお、存在しないIDを入力した場合、「該当する企業データが見つかりませんでした」というエラーメッセージが表示される。



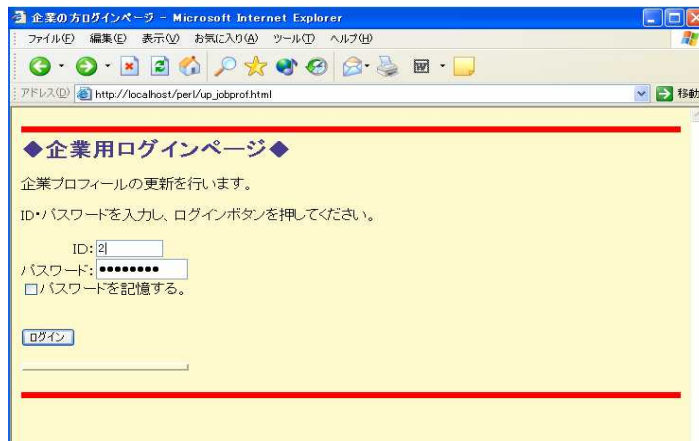
該当する企業データが見つかりませんでした。

IDは存在するが、パスワードが一致しない場合は「パスワードが違います」というエラーメッセージが表示される。

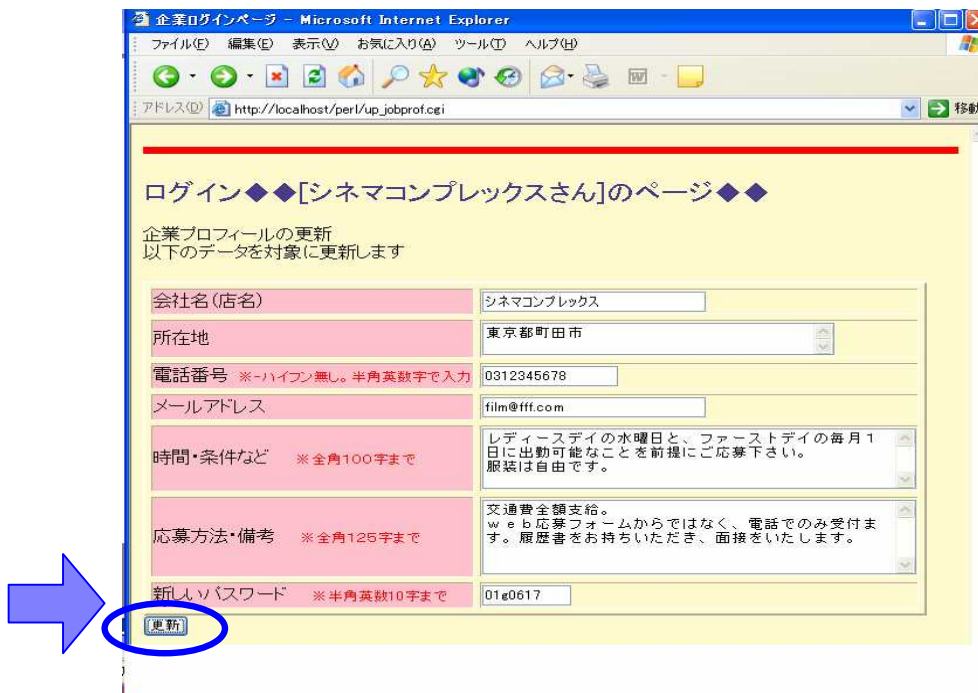


(3) 企業情報の変更・更新

ログインに成功すると、各企業の元の企業情報がデフォルトで入っている変更・更新フォームが表示される。



変更したい部分を消して新しい情報を入力し、更新ボタンをおす。



更新された内容とともに、更新完了画面が表示される。

ID	00000000002
会社名(店名)	シネマコンプレックス
所在地	東京都町田市
電話番号	0312345678
メールアドレス	film@fff.com
時間・条件など	レディースデイの水曜日と、ファーストデイの毎月1日に出勤可能なことを前提にご応募下さい。服装は自由です。
応募方法・備考	交通費全額支給。web応募フォームからではなく、電話でのみ受付ます。履歴書をお持ちいただき、面接をいたします。
パスワード	01g0617

(4) 求人情報の追加

ログイン後、求人情報追加用画面が表示される。

入力フォームに求人情報を入力する。

◆◆ログイン画面◆◆ データの追加

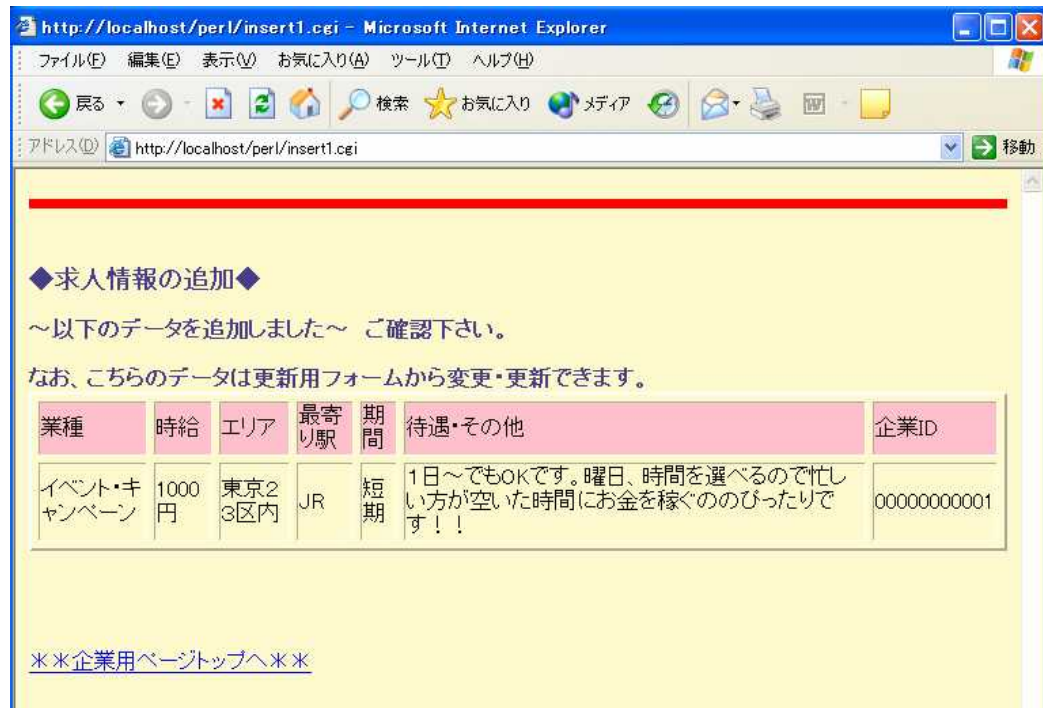
追加するデータを入力し、追加ボタンをおして下さい。

ID	00000000001
業種	イベント・キャンペーン
時給	1000 円 ※半角数字を入力
エリア	東京23区内
最寄り駅	JR山手線各駅
期間	<input checked="" type="radio"/> 短期 <input type="radio"/> 長期
待遇・その他	1日～でもOKです。曜日、時間を選べるので忙しい方が空いた時間にお金を稼ぐののひったりです!!

追加

追加ボタンをおす。

正しく入力されていれば、正常にデータは追加され、追加情報のデータとともに、追加完了画面が表示される。



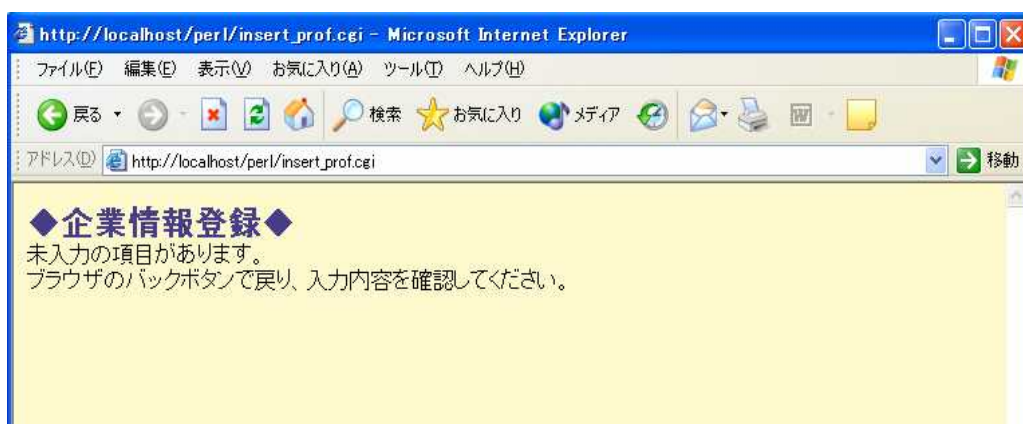
The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://localhost/perl/insert1.cgi`. The page content is as follows:

◆求人情報の追加◆
～以下のデータを追加しました～ ご確認下さい。
なお、こちらのデータは更新用フォームから変更・更新できます。

業種	時給	エリア	最寄り駅	期間	待遇・その他	企業ID
イベント・キャンペーン	1000円	東京23区内	JR	短期	1日～でもOKです。曜日、時間を選べるので忙しい方が空いた時間にお金を稼ぐののびったりです！！	00000000001

◆◆企業用ページトップへ◆◆

入力漏れがあると、「未入力の項目があります」といったエラーメッセージが表示されるので、ブラウザのバックボタンで戻って確認し、再入力する。



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://localhost/perl/insert_prof.cgi`. The page content is as follows:

◆企業情報登録◆
未入力の項目があります。
ブラウザのバックボタンで戻り、入力内容を確認してください。

(5) 求人情報の変更・更新

ログイン後、各企業が掲載している全ての求人情報が表示される。

データ更新フォームにデフォルトで元の情報が入っているのですが、変更したい部分を削除し、新しい情報を入力する。

◆◆ログイン画面◆◆

仕事データの更新
以下のデータを対象に更新します

ID	0000000001
業種	営業・テレポ
時給	850 円 ※半角数字を入力
エリア	埼玉県
最寄り駅	入間市駅
期間	<input type="radio"/> 短期 <input type="radio"/> 長期
待遇・その他	

更新

更新ボタンをおす

正しく入力されていれば更新内容とともに、更新完了画面が表示される。

データの更新
データを更新しました

ID	0000000004
業種	教育・スクール
時給	1500
エリア	東京23区内
最寄り駅	山手線各駅
期間	短期
待遇・その他	夏休み期間の塾の講師のお仕事です。小学生～高校受験を控えた中学生まで。
企業ID	0000000001

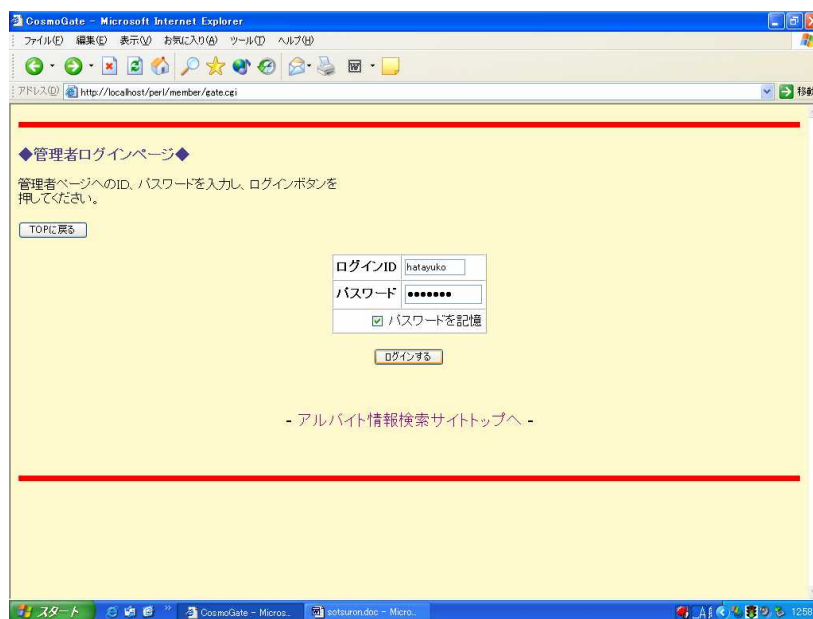
5 - 3 . 管理者用システムの流れ

管理者用システムの流れを説明する。トップページから、【管理者ページ】というリンクがあるので、そこからログイン画面へと移る。

管理者は、あらかじめ決められたID・パスワードでログインすることで、管理者用ページにアクセスできるようになっている。また、管理者は、データベース内の全てのデータ操作が可能となっている。

(1) ログイン画面

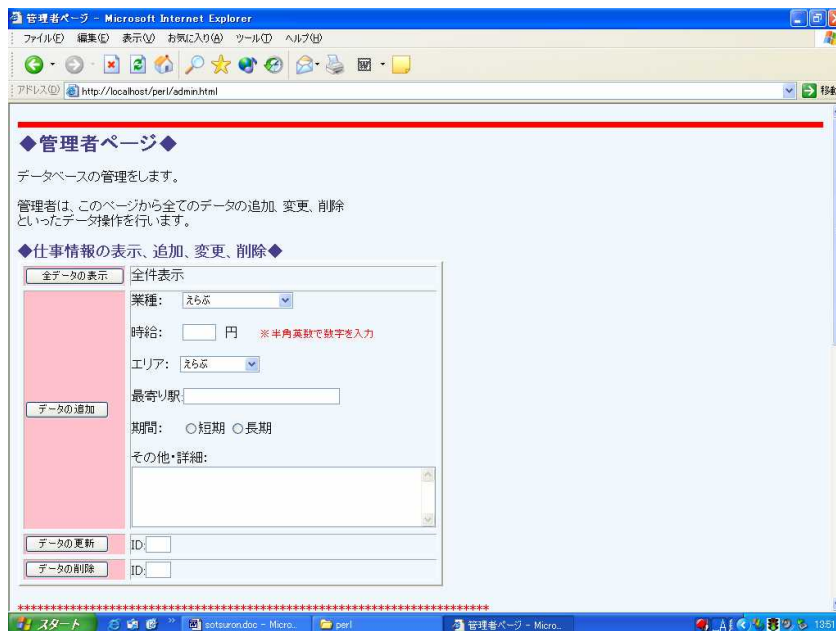
あらかじめ定められたID・パスワードを入力し、ログインボタンをおす。



ID・パスワードが一致しないと「パスワードが不正です」というエラーメッセージが表示される



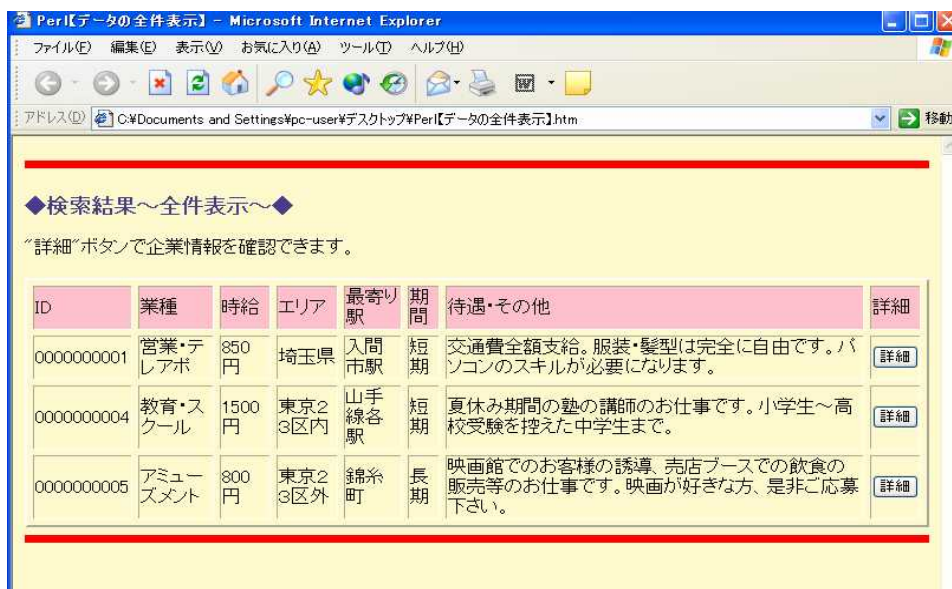
前の画面へ戻り、正しく入力されれば管理者ページが表示される。



(2) 企業情報の追加・更新・削除

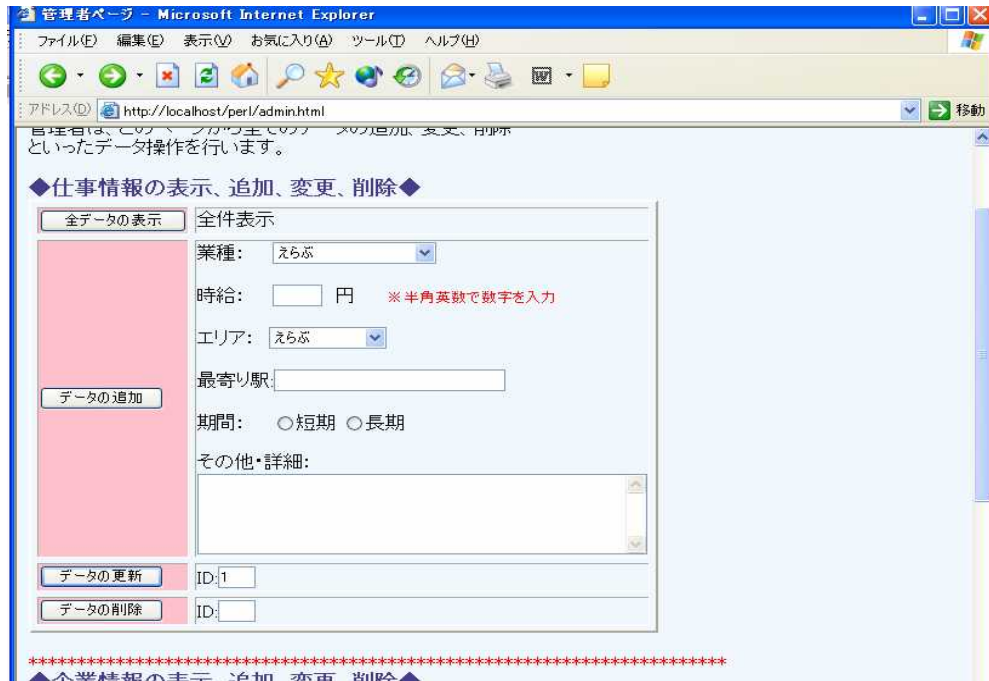
管理者用ページかのデータ操作を行う手順は企業用システムの企業情報追加と同様である。よって、画面による動作例は割愛する。

全データの表示ボタンをおすと登録された求人情報が全データ表示される

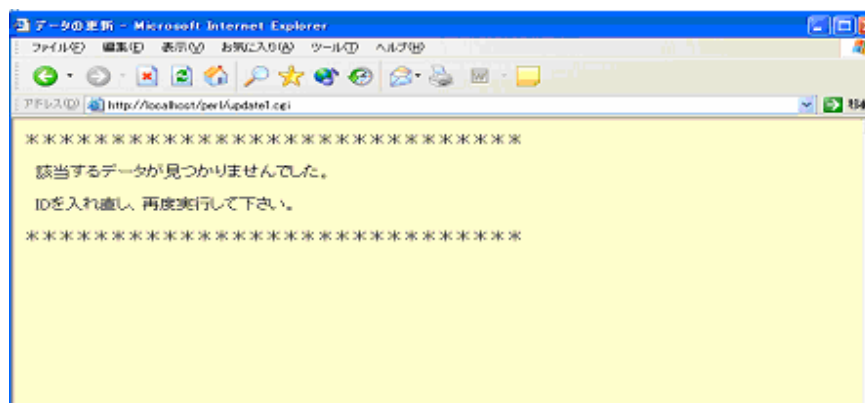


データの追加用フォームにデータを入力し、追加ボタンをおせばデータは追加される。(企業用のデータ追加機能と同様) 画面省略。

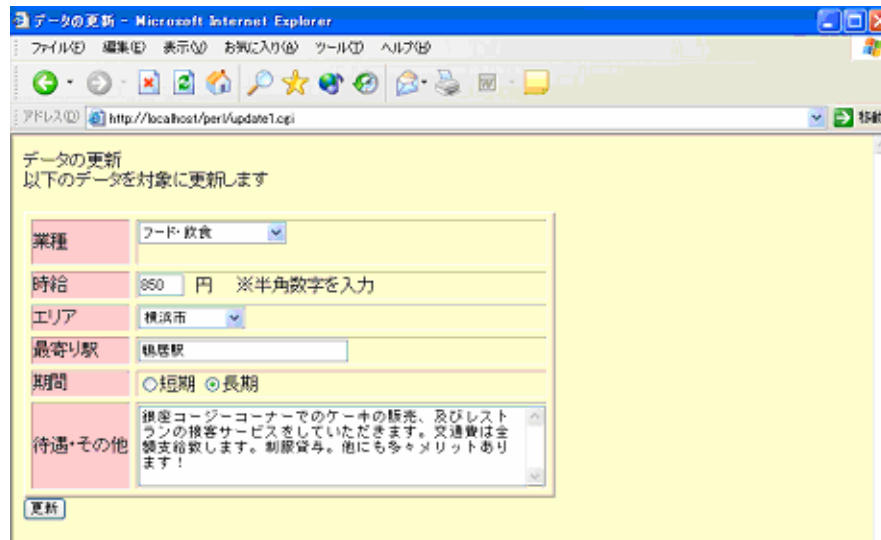
データの更新をするには、まず更新したい求人情報のID番号を入力し、更新ボタンをおす。



なお、存在しないID番号で実行すると、「該当するデータが見つかりませんでした」というエラーメッセージが表示される。

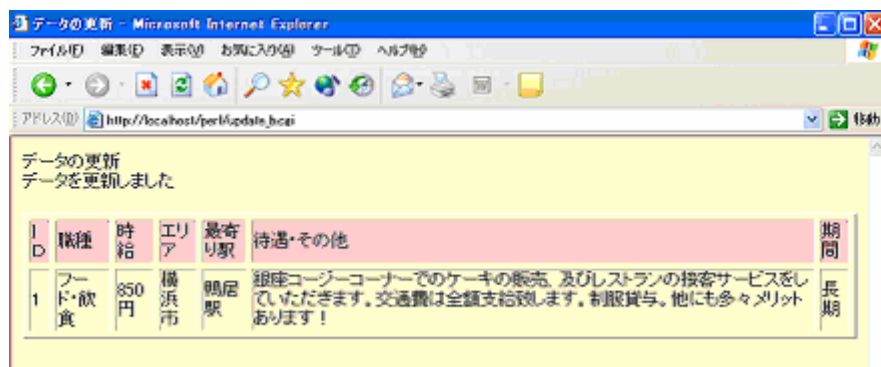


ID番号が認識されると、デフォルトで元のデータが入った更新フォームが表示される。変更したい部分を削除し、新しいデータを入力する。



更新ボタンをおす。

更新されたデータとともに、更新完了画面が表示される。



(3) 求人情報の追加・更新・削除

求人情報の追加・更新・削除は上述した「企業情報の追加・更新・削除」の操作方法、流れと同様になるので、ここでの説明は割愛することにする。

6 . プログラムの配置と解説

本項では、プログラムファイルの配置情報と、データベース処理など本システムの基本となっている部分のプログラムを紹介する。また、そのオブジェクトやメソッドについて解説する。

(1) プログラムファイルの配置

以下に本システムのプログラムファイルの配置情報を示す。

Index.html / システム全体の入り口となるトップページ

/student 学生用システムのプログラムファイル

- j_code.pl / 日本語コード変換を行うためのライブラリファイル
- Search.html / 情報検索用ページのトップ。条件などを入力するフォームの表示
 - Select.cgi / 全件表示の処理
 - Select_more.cgi / 詳細表示の処理。全件表示のプログラムから引き継いだデータ（企業ID番号）で企業情報を検索し、表示する。
 - Oubo_form.html / Web応募フォームの表示
 - F_mailer.cgi / 応募フォームのデータをメールで送信する処理
 - F_mailer_conf.pl / 送信先の指定や、メールのフォームなどを設定しているライブラリファイル
 - Confirm.html / 応募フォームの内容確認の表示
 - Thanks.html / 送信完了のメッセージの表示
- Search.cgi / 条件検索の処理。
 - Select_more.cgi / 詳細表示の処理。

/company / 企業用システムのプログラムファイル

- touroku_form.cgi / 企業の会員登録フォーム
 - insert_prof.cgi / 企業情報の登録処理
- up_jobprof.html / 企業情報更新の際のログイン処理
 - up_jobprof.cgi / 企業ごとの情報と更新用フォームの表示
 - up_jobprof_b.cgi / 更新内容のデータベース登録処理（データ更新）
- up_jobinfo.html / 求人情報更新の際のログイン処理
 - up_jobinfo.cgi / 企業ごとの求人情報と更新用フォームの表示
 - up_jobinfo_b.cgi / 求人情報のデータベース登録処理（データ更新）
- insert_jobinfo.html / 求人情報追加の際のログイン処理
 - insert_jobinfo.cgi / 求人情報の追加用フォーム表示と実行処理（データの追加）

/member / 管理者用システムのプログラムファイル

- gate.cgi / パスワード入力画面の表示とログイン処理
- admin.html / 管理者ページのトップ。データベース操作フォームの表示
 - select1.cgi / 求人情報の全件表示
 - select_more.cgi / 詳細表示
 - insert1.cgi / 求人情報の追加の処理 (データ追加)
 - update1.cgi / 指定した I D 番号の旧データと変更用フォームの表示
 - update_b.cgi / 求人情報変更の処理 (データ更新) と確認画面表示
 - delete1.cgi / 指定した I D 番号の求人情報を表示
 - delete_b.cgi / 指定して取り出した求人情報を削除 (データ削除)
 - select_prof.cgi / 企業情報の全件表示
 - insert_prof.cgi / 企業情報追加の処理 (データ追加)
 - update_prof.cgi / 指定した I D 番号の企業情報の表示と変更用フォームの表示
 - update_b.cgi / 企業情報変更の処理 (データ更新) と確認画面表示
 - delete_prof.cgi / 指定した I D 番号の企業情報を表示
 - delete_prof_b.cgi / 指定して取り出した企業情報を削除 (データ削除)

(2) オブジェクトやメソッドについて

ここでは、本システムのスクリプトの基本となっているオブジェクトやメソッドについて解説する。

< Web からデータベース操作する際のオブジェクトやメソッドについて >

まず、Web からのデータベースコントロールについて触れておく。MySQL からデータを取り出すには、「フォームから CGI が呼び出され、MySQL へ接続し、指定の DB に SQL コマンドを発行する。」この流れは、PHP や JAVA、ASP など他のスクリプトと同じである。ただ、ソースを見比べると Perl の場合は、タグの中にスクリプトを埋め込むというよりも、「タグを出力する」というイメージである。

実際に DBI/DBD を使って Perl から DB にアクセスする場合、おおまかに流れは以下ようになる。

- モジュールの読み込み
- データベース接続
- ステートメントの準備
- ステートメントの実行

{データの取り出し}
データベース切断

use DBI; /モジュールの読み込み

\$query = “実行する SQL 文 ”; /\$query に実行させたい SQL 文を格納

\$dsn = “データソースの指定”;

\$dbh = DBI -> connect (\$dsn, 'ユーザー名', 'パスワード');
/データベースハンドルを生成

データベースハンドルとは、データベースとの接続を表すオブジェクトである。上記のようにして接続し、connect はデータベースハンドルオブジェクトへのリファレンスを返す。以後は\$dbh を使って DBI へアクセスする。

\$sth = \$dbh -> prepare (\$query);
/実行する SQL 文でステートメントハンドルを生成

ステートメントハンドルとは、準備された特定の SQL 文を表すオブジェクトである。\$query に格納した SQL 文を準備する。prepare は、ステートメントハンドルオブジェクトへのリファレンスを返す。

\$sth -> execute ();
/準備されたステートメントを実行(execute)することで、SQL 文が実行される

\$rows = \$sth -> fetchrow_array; /行数を取得

fetchrow_array メソッドを発行することで、次のデータ行を取り出し、フィールドの値を持った配列を返す。

\$dbh -> disconnect(); /データベースハンドルを開放

<その他、Perl スクリプトのメソッドについて>

#! C:/Perl/bin/Perl まず、1行目にパスを必ず書かなくてはならない。パスが間違っているとプログラムは動かないし、パスの前後、途中には空白を入れてはいけない。

パスとは...CGI が呼び出された際に CGI ファイルを処理するプログラム (Perl エンジン) がどこにあるかを示すもの。

Content-type: text/html

Content-type ヘッダ...CGI プログラムからサーバ(ブラウザ)に実行結果を送るときに、それがどのような種類のデータであるかを教えるための情報である。「:」で区切ってこれに続くデータの形式を指定。HTML データの場合は「**text/html**」である。

require 'jcode.pl';

require 関数...ライブラリを読み込む。この場合、jcode.pl というライブラリに書かれているサブルーチン定義がここに書かれているのと同じ事になる。スクリプトがすっきりする。

\$method = \$ENV {'REQUEST_METHOD'}; /環境変数で送信方式を取得

read (STDIN, \$buffer, \$ENV {'CONTENT_LENGTH'});

/read 関数でデータを読み込む。環境変数 **CONTENT_LENGTH** の長さのデータを標準入力から読み込み、変数 \$buffer に格納。

フォームに入力されたデータを wwwサーバ経由で CGI プログラムに送る主な方法 (request メソッド) には GET と POST の 2 方式がある。どちらを使うかは <FORM> の **method** 属性で指定する。 Ex) <FORM method="POST" action" ">

GET...入力データでは1行にまとめられてwwwサーバによって環境変数「QUERY_STRING」に格納される。ただし、環境変数に格納できる文字列の長さには制限がある。

POST...入力データは1行にまとめられてwwwサーバによって標準入力からCGIプログラムに与えられる。CGIプログラム方では標準入力からデータを読み取るようにプログラムしておく。データの長さに制限がなく、データのバイト数が環境変数「CONTENT_LENGTH」に代入される。

環境変数...プログラム間(サーバとプログラム)でデータを共有(代入、参照)するための橋渡し役の変数。Perlの場合は連想配列%ENVに代入されるのでこの環境変数を取り出すためにはキーに環境変数名を使う。
つまり ENV{'環境変数名'} という形式で取り出すということになる。

@PAIRS = split('区切り文字',文字列); /split 関数で分解して配列に格納

split 関数...送信された文字列の中を左から検索(パターンマッチ)して区切り文字を発見するとそこで文字列を分解する。分割された文字列は左辺に格納される。

foreach \$i (@PAIRS);

/リストの要素を順次*\$i*に格納し、@PAIRSの要素がなくなるまでループする。

Foreach...区理解しを行う制御構造(つまりループ)である。リスト値を順に変数にセットしながらブロック内の文を繰り返し、リストがなくなったらループを終了する。

< プログラムの一部解説 >

up_jobinfo.cgi

//環境変数で送信方式を取得

```
$method = $ENV {'REQUEST_METHOD'};
```

```
if($method eq 'POST')
```

```
{
```

//送信方式がPOSTの場合

```
    read (STDIN, $buffer, $ENV {'CONTENT_LENGTH'});
```

//read 関数でデータを読み込み、データを変数\$buffer に格納

```
}
```

```
else
```

```
{
```

```
    $buffer = $ENV {'QUERY_STRING'};
```

```
}
```

//送信されたデータを区切り文字で分割し、変数\$key と\$val に格納。以下、送信データの中に含まれる改行コードをUNIX系OS用の¥n に変換。Jcode.pl で送信データに日本語が含まれる場合、漢字コードを sjis に統一する。\$val の値を\$key をキーにして連想配列 FORM に格納。

```
@PAIRS = split('&', $buffer);
```

```
foreach $i (@PAIRS){
```

```
    ($key,$val) = split('=', $i);
```

```
    $val =~ s/¥r¥n/¥n/g;
```

```
    $key =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("C",hex($1))/ge;
```

```
    $val =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("C",hex($1))/ge;
```

```
    jcode'convert(*$key,'sjis');
```

```
    jcode'convert(*$val,'sjis');
```

```
    $FORM{$key} = $val;
```

```
}
```

//連想配列に格納されたフォーム部品名の値をIDとパスワードの変数にそれぞれ格納する。

```
$form_id = $FORM{"form_id"};
```

```
$form_pass = $FORM{"form_pass"};
```

//省略

//

//もし、パスワードとIDどちらかでも入力されていなかったら

```
if($form_id eq "" || $form_pass eq ""){  
    print "ID・パスワードを入力して下さい。¥n";  
}
```

else

```
{
```

//ログインフォームで入力されたIDとパスワードのデータを受け取れたら、データベースに接続しIDが一致するレコード(行)をテーブルから取り出す。それを配列rowに格納し、同時に行数を取得する。

```
$dsn = "DBI:mysql:jobsearch_db:localhost";  
$dbh = DBI->connect($dsn,'chef','secret');
```

```
$query = "select count(*) from jobprof_t where id_c = ¥"$form_id ¥";  
$sth = $dbh->prepare($query);  
$sth->execute();
```

```
@rows = $sth->fetchrow_array;  
$row = $rows[0];
```

//ID番号が得られなかったら、そのテーブルにそのIDをもつ企業が存在しないということ。

```
if($row == 0)  
{  
    print "該当する企業データが見つかりませんでした。";  
}
```

else

```
{
```

```
    $query = "select * from jobprof_t where id_c = ¥"$form_id ¥";
```

//フォームから取得したID番号でテーブルを検索するSQL文を\$query
//へ格納します。

```
$sth = $dbh->prepare($query);  
$sth->execute();  
@row = $sth->fetchrow;
```

//レコードから得た、パスワードを格納しているフィールドの値とフォームから取得したデータ(パスワード)の値が一致したら、ログインということで、データの更新フォームを表示する。

```
    if($row[7] eq $form_pass)
    {
//省略・・・
```

//企業ID番号をキーとして、求人情報を格納しているテーブルの参照用ID番号と一致するフィールドを検索する。

```
    $query = "select * from jobinfo_t where id_prof = ¥"$form_id¥";
```

```
        $sth = $dbh->prepare($query);
```

```
        $sth->execute();
```

//一致するものがなくなるまで繰り返しレコードを検出する。

```
        while(@row = $sth->fetchrow_array){
```

```
print <<"HTML";
```

```
<form action="up_jobinfo_b.cgi" method="POST">
```

```
<table border="2" cellspacing="5">
```

```
<tr>
```

```
<td bgcolor="pink"> I D </td>
```

```
<td>$row[0]</td>
```

```
</tr>
```

//データ更新用フォームに検出したデータを表示させる。

```
<tr>
```

```
    <td bgcolor="pink">業種</td>
```

```
    <td><select name="update_shokushu">
```

```
        <option value="$row[1]">$row[1]
```

```
            <option value="フード・飲食">フード・飲食
```

```
            <option value="販売・ファッション">販売・ファッション
```

```
        <option value="アミューズメント">アミューズメント
```

```
        <option value="イベント・キャンペーン">イベント・キャンペーン
```

```
        <option value="受付・事務">受付・事務
```

```
            <option value="営業・テレアポ">営業・テレアポ
```

```
            <option value="教育・スクール">教育・スクール
```

```

        <option value="配送・物流">配送・物流
        <option value="IT・クリエイティブ">IT・クリエイティブ
        <option value="その他サービス">その他サービス
    </select><br><br>
</td>
</tr>
<tr>
    <td bgcolor="pink">時給</td>
    <td><input type="text" name="update_money" size="5" value="$row[2]"> 円
    半角数字を入力
</td>
</tr>
<tr>
    <td bgcolor="pink">エリア</td>
    <td><select name="update_area">
        <option value="$row[3]">$row[3]
            <option value="東京 2 3 区内">東京 2 3 区内
            <option value="東京 2 3 区外">東京 2 3 区外
        <option value="横浜市">横浜市
        <option value="他・神奈川">他・神奈川
        <option value="埼玉県">埼玉県
            <option value="千葉県">千葉県
    </select>
</td>
</tr>
<tr>
    <td bgcolor="pink">最寄り駅</td>
    <td><input type="text" name="update_station" size="35" value="$row[4]">
</td>
</tr>
<tr>
    <td bgcolor="pink">期間</td>
    <td><input type="radio" name="update_period" value="短期">短期
        <input type="radio" name="update_period" value="長期">長期
</td>
</tr>
<tr>
    <td bgcolor="pink">待遇・その他</td>

```

```
<td><textarea      NAME="update_detail"      rows="5"      cols="50"
value="$row[6]"></textarea>
</td>
</tr>
```

//更新する企業 ID 番号のデータを入力フォームの初期値として表示する。入力フォームの内容（ID 番号は下記 HTML 部分の隠しフォームで）は送信データとして update_jobinfo.cgi に渡される

```
</table>
<input type="hidden" name="update_id" value="$row[0]">
<input type="hidden" name="id_prof" value="$row[7]">
<input type="submit" value="更新">
</form>
```

HTML

```
}
```

```
}
```

```
else{
```

```
//検出したフィールドのデータとフォームより取得したデータが一致しなかったら
```

```
print "パスワードが違います。";
```

```
}
```

```
}
```

```
//データベースハンドルを開放
```

```
$dbh->disconnect();
```

```
}
```

```
print <<"HTML";
```

```
</body>
```

```
</html>
```

```
HTML
```

```
exit;
```

up_jobinfo_b.cgi

//省略

```
use DBI;
```

```
require "./jcode.pl";
```

```
$method = $ENV{'REQUEST_METHOD'};
```

```
if($method eq 'POST')
```

```
{
```

```
    read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
```

```
}
```

```
else
```

```
{
```

```
    $buffer = $ENV{'QUERY_STRING'};
```

```
}
```

```
@PAIRS = split('&', $buffer);
```

```
foreach $i (@PAIRS)
```

```
{
```

```
    ($key,$val) = split('=', $i);
```

```
    $val =~ s/¥r¥n/¥n/g;
```

```
    $key =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("C",hex($1))/ge;
```

```
    $val =~ s/%([0-9a-fA-F][0-9a-fA-F])/pack("C",hex($1))/ge;
```

```
    jcode'convert(*$key,'sjis');
```

```
    jcode'convert(*$val,'sjis');
```

```
    $FORM{$key} = $val;
```

```
}
```

//連想配列 FORM に格納された値をそれぞれ変数に格納

```
$update_id = $FORM{"update_id"};
```

```
$update_shokushu = $FORM{"update_shokushu"};
```

```
$update_money = $FORM{"update_money"};
```

```
$update_area = $FORM{"update_area"};
```

```
$update_station = $FORM{"update_station"};
```

```
$update_period = $FORM{"update_period"};
```

```
$update_detail = $FORM{"update_detail"};
$id_prof = $FORM{"id_prof"};
```

//フォームから取得した要素でテーブルを更新する SQL 文を\$query へ格納する。

```
$query = "update jobinfo_t set shokushu_c = ¥'$update_shokushu¥',
money_c = ¥'$update_money¥', area_c = ¥'$update_area¥',
station_c = ¥'$update_station¥', period_c = ¥'$update_period¥',
detail_c = ¥'$update_detail¥', id_prof = ¥'$id_prof¥' where id_c =
¥'$update_id¥'";
```

```
$dsn = "DBI:mysql:jobsearch_db:localhost";
$dbh = DBI->connect($dsn,'chef','secret');
$stmt = $dbh->prepare($query);
$stmt->execute();
$dbh->disconnect();
```

```
print "Content-type:text/html¥n¥n";
print <<"HTML";
<html>
<head>
<meta http-equiv="content-type"
content="text/html;charset=shift_JIS">
<title>データの更新</title>
</head>
<body bgcolor="#fffacd" text="#000000">
```

//更新した ID 番号のデータを表示する。

```
<p>データの更¥新<br>データを更¥新しました</p>
<table border="2" cellspacing="5">
<tr>
<td bgcolor="pink"> I D</td>
<td>$update_id</td>
</tr>
<tr>
<td bgcolor="pink">業種</td>
```

```
<td>$update_shokushu</td>
</tr>
<tr>
  <td bgcolor="pink">時給</td>
  <td>$update_money</td>
</tr>
<tr>
  <td bgcolor="pink">エリア</td>
  <td>$update_area</td>
</tr>
<td bgcolor="pink">最寄り駅</td>
  <td>$update_station</td>
</tr>
<td bgcolor="pink">期間</td>
  <td>$update_period</td>
</tr>
<tr>
  <td bgcolor="pink">待遇・その他</td>
  <td>$update_detail</td>
</tr>
<tr>
  <td bgcolor="pink">企業 I D</td>
  <td>$id_prof</td>
</tr>
</table>
</body>
</html>
HTML
exit;
```

7. 考察・課題

本システムは、情報の掲載から学生の情報検索・応募、また情報の管理までをWeb上で一括してできる機能を備え、実際に動かせるようになった。これによって、当初の目的であった、あらゆる手間や時間の削減を実現し、また情報管理や時間的ギャップなどの問題解決に向けてかなり前進した。

ただ、これで全てが解決されたというわけではない。まだまだ未完成のこのシステムには検討すべき課題が多く残される。特に管理者用システムに関しては、情報の監視・管理をしやすいような改善が必要である。

会員登録時の処理の自動化

登録完了、ID・パスワードの確認メールは管理者によってなされることになっている。しかし、今後登録企業が多くなればなるほどこの作業が大きな負担になると考えられる。よって、この確認メール送信を自動化することが必要になる。

また、新しく企業が登録された際にメールか何かでそれを管理者に知らせるメール機能をもたせることが望まれる。

情報の管理機能

学生の利用状況や企業の情報更新状況の記録をフォームで見られるような、本システムの監視に関する機能を備え、より質の高い管理体制を整えたい。

インターフェースの工夫・改善

より見やすく、より利用しやすいシステムにするため、インターフェースの改善をしていきたい。特に、検索結果の表示に関して、今では全ての情報が1ページに表示されるようになっている。これでは利用者はページをスクロールしなくてはならなくなるので、ページ更新機能を設定する必要がある。

セキュリティに関して

本システムでは私自信の知識不足もあり、セキュリティに関してはほとんど考慮されていない。しかし、Web応募などで個人情報も扱うようになっているので、実用するためにはセキュリティの強化が必須である。

8 . 謝辞

最後に、本システムの作成にあたり、熱心なご指導、ご支援して下さいました山本昌弘教授、重定如彦先生をはじめ、ご協力頂いた同期ゼミ生の皆様に心より感謝申し上げます。

9 . 参考文献

「一週間でマスターする CGI for Windows」

福島 靖浩 (著)

MYCOM

「標準 MySQL」

田中ナルミ / 阿部忠光 (著)

株式会社ソフトウェアエンシー 監修

SOFTBANK

「Q&A 100で学ぶ! Perl / CGIプログラミング」

森下 幸治 (著)

エーアイ出版