问题一：MIME TYPE 写的太冗余，或者是直接根据文件名来判断 header 的内容：

```
        var pathname = url.parse(request.url).pathname;
        if (pathname == "/") {
        fs.readFile('./index.html', function(err, html) {...
        });
    }
    if (pathname == "/index2.css") {...
    }
    if (pathname == "/index.css") {...
    }
    if (pathname == "/jQuery.js") {...
    }
    if (pathname == "/index.js") {
        fs.readFile('./index.js', function(err, js) {
            response.writeHeader(200, {
                "Content-Type": "text/javascript"
            });
            response.write(js);
            response.end();
        });
    }
    if (pathname=="/login") {...
```

sol. 使用 querystring 解析 url，然后根据请求文件的后缀名来判断 header 类型

```
var MIME = {
    '.html': 'text/html',
    '.css': 'text/css',
    '.js': 'application/javascript'
}

fs.readFile(realPath, "binary", function (err, file) {
    if (err) {
        response.writeHead(500, { 'Content-Type': 'text/plain' });
        response.end(err);
    } else {
        response.writeHead(200, { 'Content-Type': MIME[path.extname(realPath)] });
        response.write(file, 'binary');
        response.end();
    }
});
```

问题二：硬编码到 JS 中的 HTML

```
fs.appendFile('store.txt', JSON.stringify(user) + "\n", function(err) {
    if (err) throw err;
    var newhtml = "";
    newhtml += ("<head>");
    newhtml += ("<title>new<\/title>");
    newhtml += ("<link rel=\"stylesheet\" type=\"text/css\"  href=\"index.css\">");
    newhtml += ("<\/head>");
    newhtml += ("<div><h1>Sign Up Success<\/h1><div>用户名 <span> " + user.username + "<\/span> <\/div>");
    newhtml += ("<div>学号: <span>  " + user.num + "<\/span><\/div>");
    newhtml += ("<div>电话: <span>  " + user.tell + "<\/span><\/div>");
    newhtml += ("<div>邮箱: <span>  " + user.mail + "<\/span><\/div>");
    newhtml += ("<a href=\"http:\/\/127.0.0.1:8000\">返回<\/a><\/div>");
    response.writeHeader(200, {
        "Content-Type": "text/html"
    });
    response.write(newhtml);
    response.end();
```

sol. 使用 jade/自己做模板
具体用法不多说。。自己看文档 http://jade-lang.com/api/

1. Jade 编译 HTML

```
var jade = require('../')
  , path = __dirname + '/xxx.jade'
  , str = require('fs').readFileSync(path, 'utf8')
  , fn = jade.compile(str, { filename: path, pretty: true });

console.log(fn({ '模板变量名': '变量值' }));
```

2. 自己手动 replace 做模板

```
response.writeHead(200, {"Content-Type": "text/html; charset=utf-8"});
fs.readFile("upload.html", function(error, data) {
  if (error) throw error;
  //console.log(newuser.username);
  var pagestring = data.toString();
  pagestring = pagestring.replace("{{UserName}}", newuser.username);
  pagestring = pagestring.replace("{{StudentId}}", newuser.studentId);
  pagestring = pagestring.replace("{{Tele}}", newuser.tele);
  pagestring = pagestring.replace("{{Mail}}", newuser.mailbox);
  response.write(pagestring);
  response.end();
});
```

一些小小的改进：

1. 前端校验表单之后格式非法应该立即给予提示，而不是提交的时候再给出。具体实现可以监听 onkeypress 和 onblur。
2. 用户名/邮箱判重等问题则可以使用 ajax 方式与服务器交互