MVC – Model-View-Controller pattern. Model represents an object. It can also have logic to update controller if its data changes. View represents the visualization of the data that model contains. Controller acts on both Model and view. It controls the data flow into model object and updates the view whenever data changes.
My project is blog about films. User can register and write blog as a blogger or can just read blogs.
User sends a request to controller. Then controller affects on model or controls model. Model updates the files of view. View shows information on user's interface.
In controller I did the main work.

```
String template=readFile("template.html");
	if(!path.equals("/") ) {
		path=path.substring(1);
		String sub=readFile(path);
		response=template.replace("%content%", sub);
	}else if(path.equals("/")){
		String sub=readFile("file.html");
		response = template.replace("%content%", sub);
}
```

- This piece of code shows the first page. Template is unchangeable, contents change by clicking on links on the template.

```
if(path.equals("/reg")){
	InputStream is = t.getRequestBody();
	BufferedReader reader = new BufferedReader(new InputStreamReader(is));

	Vector<String> data = new Vector<>();
	String l;
	l = reader.readLine();
	StringTokenizer st = new StringTokenizer(l, "=&");
	while (st.hasMoreTokens()) {
		data.add(st.nextToken());
	}
		return data;
}
```

- Registration. Here I wrote the code for receiving data from forms in html files. It receives data from form and divides data into name and the value. Then saves it to vector of string.

In model package, I have classes for Database and User. After the registration, information about user will save in database. View package contains html files.