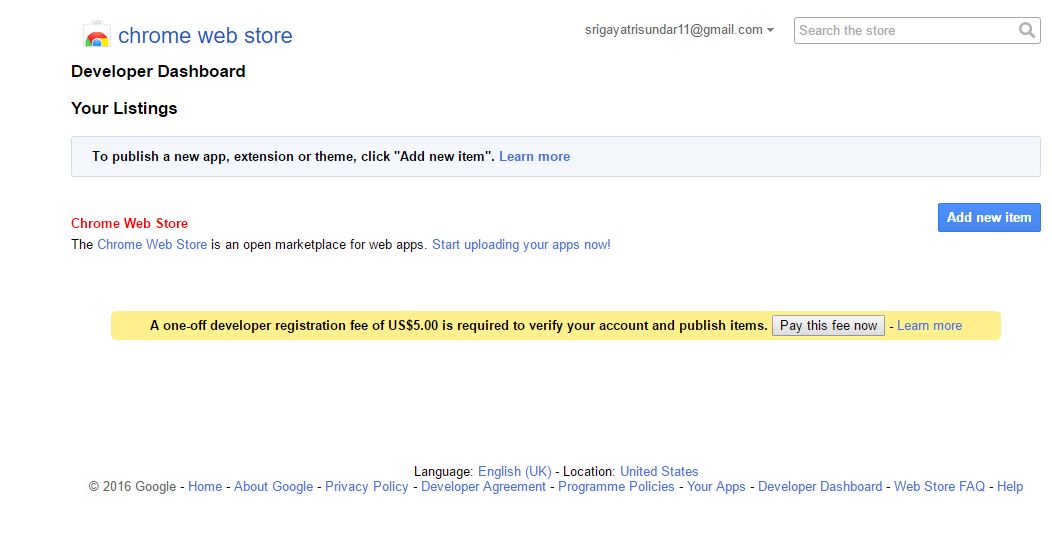
**Deployment Plan**

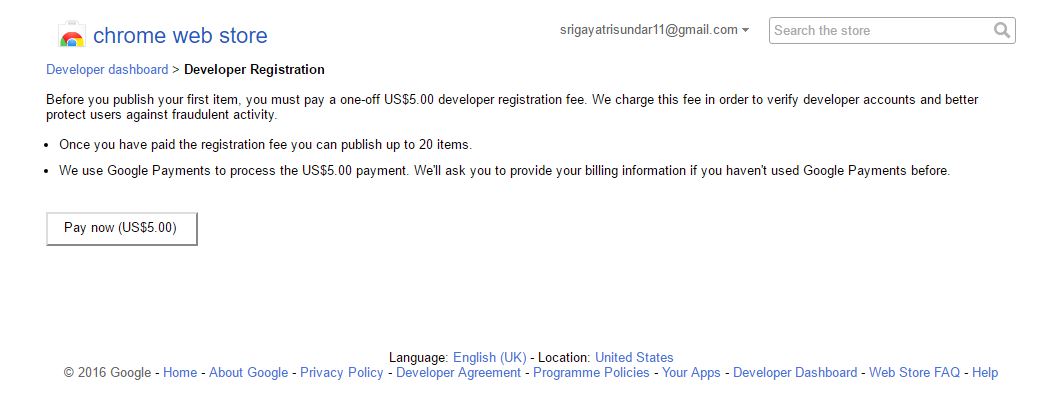
**Slacker Hacker**

Our product is called ‘Slacker Hacker’. It is a Google Chrome extension that tracks your study time and blocks you from websites of your choice. It also makes statistically analyzed graphs of your productive and non-productive time in a day and allows you to share it on Facebook.

Since it is a Google Chrome extension, our developing team (team 9) are planning to make it live by uploading it to the Google chrome web store.

Our deployment plan to upload it on the Chrome Web Store is pretty simple.

1. We first will create a developer account for the extension instead of using our own personal account.
2. Then upload our extension’s zip folder that contains our app name and Version number in the web store dashboard.
3. Our extension is going to be free of charge, so we don’t have to pick a payment system like Google checkout merchant account etc.
4. Provide store content for our extension that isn’t in our zip file like a detailed description of the extension. Screenshots, videos, links and related sites.
5. Pay the developer signup fee. Before we publish our extension, we are entitled to pay a one-time $5 developer signup fee.



1. Get app constraints such as an extension ID etc
2. We are also planning to first publish the extension to test accounts before we make it live
3. When the extension passes the beta version and gives successful test account results, we publish it to the world!
4. Since our extension is education related, we would also like to publish this on Google Play for Education. Google Play for Education is a separate store that lets educators find, buy and deploy apps to their students with just a few clicks

Google Play for Education has its own set of instructions in order to publish our extension.

1. We first have to register for a publisher account
2. Pay $25 USD registration fee using google payments.
3. When our registration is verified, we can sign into our developer console, which is the home for our extension app publishing operations and tools on Google Play.

With the current product that we have, it allows only a single user. So before me deploy it, we would like to extend it to a multi user platform by having our own server and data storage facilities.

Out to-do list currently uses a NoSQL cloud database called firebase where data is stored as JSON, synced to all connected clients in real-time and available when the app/extension goes offline. It also has other features like Authentication and Hosting. But we are planning to shift to Amazon AWS.

Amazon AWS is a subsidiary of Amazon.com which offers a suite of cloud computing services that make up an on-demand computing platform.

The Backend for our Chrome extension is written in PHP and JavaScript. For our data Storage, we are using a MySQL database. Currently our PHP application and the MySQL data store are hosted in the KU server.

To put our extension into production for the millions of chrome users around the world is not an easy task. To make things easier we have planned to leverage the advantage of cloud computing. It allows us to quickly and cheaply get our application online. The following is our plan for production.

Since we require fine grained control of our servers, we have decided to use Amazon Web Services (AWS) as our infrastructure as a service (IaaS) provider. We will request an Elastic Cloud Compute (EC2) instance from amazon to run our server.

AWS is an IaaS provider, which means we completely eliminate our ‘Startup Cost’ by not requiring us to invest in our own hardware. We can ask specify our requirements for resources such as ‘Power Processing’, ‘Storage’, ‘RAM’ and network bandwidth. This decreases the need for hardware resources almost entirely.

AWS also offers us a ‘Pay as you go’ policy which only requires us to pay for what we use and give us access to nearly unlimited resources as and when required.

Since our application will be access by thousands of users, we must make sure that we have enough resources to handle periods of high traffic. AWS offers us automatic scaling to our application, which means that storage, processing power and bandwidth will be provided on demand depending on requirements and traffic. This will help us save a lot of money buying extra hardware to handle periods of high traffic.

To attend the Google I/O conference, the registration fee is $300 per head. Adding all the flight charges, stay, and food and setting up booths etc., the total charge will come up to $1000 to $1200 per head.

We also would love to attend the Google Chrome Dev Summit. It is an exclusive event to connect with chrome engineers and leading web developers for a two-day exploration of building beautiful and fast mobile web applications. This would be a great opportunity for us to make connections and raise our horizons.